GEORGIA INSTITUTE OF TECHNOLOGY OFFICE OF CONTRACT ADMINISTRATION

[x] ORIGINAL
[ ] REVISION NO. ____

PROJECT ADMINISTRATION DATA SHEET

Project No. A-3008 DATE: 7/27/81

Project Director: B. B. Wise ~~School~~/Lab ECSL/CCB

Sponsor: Naval Civil Engineering Laboratory, Port Hueneme, CA

Type Agreement: Delivery Order No. F33657-80-G-0077 ZZ01 (Under BOA)

Award Period: From 7/6/81 To 1/1/82 (Performance) 1/15/82 (Reports)

Sponsor Amount: $80,039 Contracted through:

Cost Sharing: N/A GTRI/~~GIT~~

Title: Man/Machine Interface Device

ADMINISTRATIVE DATA OCA CONTACT Faith G. Costello

1) Sponsor Technical Contact: Mr. Karlin Canfield, Naval Civil Engineering Laboratory Port Hueneme, CA 93043

2) Sponsor Admin./Contractual Contact: Officer in Charge of Contracts, Naval Civil Engineering Laboratory, Port Hueneme, CA 93043

Reports: See Deliverable Schedule Security Classification: N/A

Defense Priority Rating: DO C-2 under DMS Reg. 1

RESTRICTIONS

See Attached Gov't Supplemental Information Sheet for Additional Requirements.

Travel: Foreign travel must have prior approval - Contact OCA in each case. Domestic travel requires sponsor approval where total will exceed greater of $500 or 125% of approved proposal budget category.

Equipment: Title vests with Gov't; except that items costing less than $1,000 vest with GIT upon acquisition; if prior approval to purchase is obtained from Contracting Officer

COMMENTS:

COPIES TO:

Administrative Coordinator	Research Security Services	EES Research Public Relations (
Research Property Management	~~Reports~~ Coordinator (OCA)	Project File (OCA)
Accounting Office	Legal Services (OCA)	Other:
Procurement/EES Supply Services	Library, Technical Reports	

GEORGIA INSTITUTE OF TECHNOLOGY OFFICE OF CONTRACT ADMINISTRATION

SPONSORED PROJECT TERMINATION SHEET

Date 6/17/82

Project Title: Man/Machine Interface Device

Project No: A-3008

Project Director: B. B. Wise

Sponsor: Naval Civil Eng. Lab

Effective Termination Date: 5/23/82

Clearance of Accounting Charges: 5/23/82

Grant/Contract Closeout Actions Remaining:

- [x] Final Invoice ~~and Closing Documents~~
- [ ] Final Fiscal Report
- [ ] Final Report of Inventions
- [x] Govt. Property Inventory & Related Certificate
- [ ] Classified Material Certificate
- [ ] Other ______

Assigned to: ECSL/CCB (~~School~~/Laboratory)

COPIES TO:

RAN ~~Administrative Coordinator~~	Research Security Services	EES Public Relations (2)
Research Property Management	Reports Coordinator (OCA)	Computer Input
Accounting	Legal Services (OCA)	Project File
Procurement/EES Supply Services	Library	Other ______

FORM OCA 10:781

A-3008

Georgia Institute of Technology

ENGINEERING EXPERIMENT STATION

ATLANTA, GEORGIA 30332

Mr. K. J. Canfield
Navy Civil Engineering Laboratory
Port Hueneme, Ca 93043

Subject: Delivery Order No. F33657-80-G-0077 ZZ01 (A-3008)
Man/Machine Interface Device

Dear Mr. Canfield:

This is the first of a series of progress reports addressing the status of the subject contract. This report covers the period 20 July 1981 to 15 August 1981.

SCHEDULE ADJUSTMENT

1. Contract paperwork was not received at Georgia Tech until 20 July 1981, thus work could not commence until that date. By mutual agreement, the work start date for purposes of contract deliverable schedule will be 20 July vice 6 July 1981.

2. In order to facilitate manpower scheduling and to assure a proper and complete software design, it is requested that the proposed design report delivery date be extended to 16 September 1981.

3. To accommodate scheduled holiday periods occurring near the end of this project, it is suggested that the device demonstration be slated for 5 January 1982. This would result in a final deliverable date of 26 January 1982.

4. Unless otherwise directed, a request for a no-cost modification to the contract, incorporating the schedule changes detailed above, will be submitted.

STATUS

1. Computer graphics hardware has been ordered. This early order was accomplished to take advantage of the current 1-year warranty. Any orders received after 1 August 1981 would have carried only a 90-day warranty.

2. Top level operator/interface interactions and screen layout have been finalized in preparation for the software program design phase.

PLAN

Software program design and design report are scheduled for completion during the following month.

FUNDING STATUS

Initial expenditure reports on new project not yet available. Data will be included in subsequent reports.

Sincerely,

Billy B. Wise
Project Director
Command and Control Programs
Electronics and Computer Systems Laboratory

Approved

H. Bennett Teates
Head
Command and Control Programs
Electronics and Computer Systems Laboratory

Georgia Institute of Technology
ENGINEERING EXPERIMENT STATION
ATLANTA, GEORGIA 30332

September 21, 1981

OICC
Naval Civil Engineering Center
Port Hueneme, California 93043

SUBJECT: Deliverable No. 3, September Letter Progress Report (A-3008)

REFERENCE: Contract No. F33657-80-G-0077 ZZ01 - Man-Machine Interface Device

Dear Sir:

This is the second of a series of progress reports addressing the status of the referenced contract. This report covers the period August 16 to September 18, 1981.

STATUS

1. Computer graphics equipment has been delivered, set-up, and is working as advertised. An additional piece of equipment, a hardware real-time clock ($150), has been ordered for the Chromatics.

2. Output printer is due for delivery in about thirty (30) more days. A suitable Georgia Tech printer is available for use in the interim, thus no delay in progress will result from late delivery.

3. The Carroll touch panel kit has been delivered, assembled and tested. Two bad components were discovered, and replacements have been ordered from the factory under terms of the warranty. No program delays are anticipated as a result of this problem.

4. An interim software design report has been completed and forwarded via appropriate channels.

PLAN

In anticipation of only minor changes and/or additions by the technical sponsor to the interim software design, implementation of the proposed design has commenced in accordance with the plan.

PROBLEM AREAS

None

FUNDING STATUS

Full Contract Budget	$80,039.00
less dollars expended or encumbered to date	- 38,270.24
Balance as of 8/31/81	$41,768.76

Yours truly,

Billy B. Wise
Project Director,
Command and Control Programs
Electronics & Computer Systems
Laboratory

Approved:

H. Bennett Teates
Head,
Command and Control Programs
Electronics & Computer Systems
Laboratory

BBW:dar

cc: F.L. Cain
C.B. Mahaffey
H.B. Teates
Office of Contract Administration
File - A-3008

/ - - '

Georgia Institute of Technology
ENGINEERING EXPERIMENT STATION
ATLANTA, GEORGIA 30332

October 20, 1981

OICC
Naval Civil Engineering Center
Port Hueneme, California 93043

SUBJECT: Deliverable No.4, October Letter Progress Report (A-3008)

REFERENCE: Contract No. F33657-80-G-007 ZZ01
Man-Machine Interface Device

Dear Sir:

This is the third of a series of progress reports addressing the status of the referenced contract. This report covers the period September 19 to October 19, 1981.

STATUS

1. The Chromatics Real-Time clock has been received and installed.

2. The Carroll touch panel was returned to the factory for warranty repair. It has subsequently been reinstalled and is operating as advertised.

3. The supplier reports that the NEC printer should be delivered by the end of October. The simplified keyboard has been ordered.

4. The technical monitor requested one change to the software design, to accommodate EMCS report presentation on either the CRT display or on the hard copy printer. This change has been incorporated into the design.

5. Major work efforts in the last month were directed towards development and testing of concepts for emulating a real-time system with the Chromatics, and coding software modules from the design document.

PLAN

Effort in the next month will be directed toward implementation of the real-time emulation concepts developed last month, and coding of additional software modules. As with every software development project, there is a degree of uncertainty with regard to the ultimate size of the program. Two contingency options are being investigated for applicability should the program size exceed the available 32K of user RAM in the Chromatics. One option is to segment the program and call subroutines from the disc when needed. This would result in little extra cost, but would very likely slow down the operation of the MMI device to the point where it would bear little or no resemblence to the real-time interface being simulated. The second option, and the recommended one, is to acquire the Chromatics CP/M operating system. This option would cost an additional $3350, however, it would provide 62K of user RAM and thereby assure actual real-time operation speed of the MMI device by accommodating the entire software program in main memory. Should this situation arise, sponsor concurrence will be obtained prior to implementing either option.

PROBLEM AREAS

None

FUNDING STATUS

Full Contract Budget	$80,039.00
less dollars expended or encumbered to date	-46,423.14
Balance as of 9/30/81	$33,615.86

Yours trulv.

Billy B. Wise
Project Director
Command and Control Programs
Electronics & Computer Systems Laboratory

Approved:

H. Bennett Teates
Head
Command and Control Programs
Electronics & Computer Systems Laboratory

cc: F. L Cain
C. B.Mahaffey
H. B. Teates
Office of Contract Administration
File A-3008

Georgia Institute of Technology
ENGINEERING EXPERIMENT STATION
ATLANTA, GEORGIA 30332

November 20, 1981

OICC
Naval Civil Engineering Center
Port Hueneme, California 93043

SUBJECT: Deliverable Number 5, November Letter Progress Report (A-3008)

REFERENCE: Contract No. F33657-80-G-007 ZZ01
Man-Machine Interface Device

Dear Sir:

This is the fourth of a series of progress reports addressing the status of the referenced contract. This report covers the period October 20 to November 20, 1981.

STATUS

1. The simplified auxiliary keyboard has been received.

2. The NEC printer and feeders have been received and checked out.

3. Work effort in the last month has been devoted to coding additional software modules for the MMI operating program.

4. It has been determined through extrapolation from modules already coded that the software program size will be about 46K total, exceeding the 30K available memory in the current Chromatics configuration. See the Plan section of this report for action to be taken.

5. The technical monitor has inquired as to the availability of a video interface to allow multiple color terminal presentations for demonstration purposes. This capability is available from Chromatics as an option. See the PLAN section of this report for more information.

PLAN

In order to meet the need for additional main memory capacity, it is proposed that the CP/M option discussed in last month's progress report be implemented. A request is being submitted for modification of the statement of work to incorporate the CP/M operating system, acquire the video interface capability, and extend the contract completion date to the end of February, 1982. Total additional funds being requested is approximately $8,138. This request is being submitted under separate cover. In the interim, work will continue on writing software modules for the MMI program.

PROBLEM AREA

None

FUNDING STATUS

Full Contract Budget	$80,039.00
less dollars expended or encombered to date	-54,163.27
Balance as of 10/31/81	$25,875.73

Yours truly,

Billy B. Wise,
Command and Control Programs
Electronics & Computer Systems
Laboratory

Approved:

H. Bennett Teates,
Head, Command and Control Programs
Electronics & Computer Systems Laboratory

BBW/HBT:dar

A-3008

Georgia Institute of Technology

ENGINEERING EXPERIMENT STATION

ATLANTA, GEORGIA 30332

December 18, 1981

OICC
Naval Civil Engineering Center
Port Hueneme, California 93043

SUBJECT: Deliverable Number 6, December Letter Progress Report (A-3008)

REFERENCE: Contract No. F33657-80-G-007 ZZ01, Man-Machine Interface Device

Dear Sir:

This is the fifth of a series of progress reports addressing the status of the referenced contract. This report covers the period November 21 to December 18, 1981.

STATUS

Effort during this period was devoted to the writing and testing of software modules from the MMI design manual. Software development is considered to be on schedule as planned.

PLAN

Contingent upon receipt of a contract modification discussed with the technical contract monitor, it is planned to acquire the necessary additional components for the CP/M capability. With those components in hand, it will be possible to compile and begin testing of the MMI operating program.

PROBLEM AREAS

None

FUNDING STATUS

Full Contract Budget	$80,039.00
less dollars expended or encumbered to date	-58,365.92
BALANCE as of 11/30/81	$21,673.08

Sincerely,

Billy B. Wise,
Project Director
Command and Control Programs
Electronics & Computer Systems Laboratory

Approved:

for H. Bennett Teates,
Head, Command and Control Programs
Electronics & Computer Systems Laboratory

BBW/HBT:dar

+ 3008

Georgia Institute of Technology

ENGINEERING EXPERIMENT STATION

ATLANTA, GEORGIA 30332

January 27, 1982

OICC
Naval Civil Engineering Center
Port Hueneme, California 93043

Subject: Deliverable Number 7, January Progress Report (A-3008)

Reference: Contract No. F33657-80-G-007 ZZ01, Man-Machine Interface Device

Dear Sir:

This is the sixth of a series of progress reports addressing the status of the referenced contract. This report covers the period December 19, 1981, to January 20, 1982.

STATUS

Software development is in its final stages. The main graphics routine, which ties together the entire program, will be the last module to be completed. Software development is considered to be on schedule.

PLAN

The next step will be to incorporate CP/M capability into the Chromatics MMI device and proceed with final program debugging. Procurement of the additional hardware and software for CP/M is awaiting arrival of a contract modification providing additional funding for materials and extending the performance period of the contract. If the subject contract modification is not in place by January 31, 1982, there will certainly be an adverse impact on the delivery schedule and possibly on contract cost. The date for a demonstration cannot be set until the CP/M equipment has been received and installed.

PROBLEM AREAS

None

FUNDING STATUS

Full Contract Budget	$80,039.00
less dollars expended or encumbered to date	-64,327.00
Balance as of 12/20/81	$15,712.00

Sincerely,

Billy B. Wise,
Command and Control Programs
Electronics & Computer Systems
Laboratory

Approved:

H. Bennett Teates,
Head, Command and Control Programs
Electronics & Computer Systems Laboratory

BBW/HBT:dar

Georgia Institute of Technology
ENGINEERING EXPERIMENT STATION
ATLANTA, GEORGIA 30332

March 10, 1982

OICC
Naval Civil Engineering Center
Port Hueneme, California 93043

SUBJECT: Deliverable Number 8, February Letter Progress Report (A-3008)

REFERENCE: Contract No. F33657-80-G-007 ZZ01, Man-Machine Interface Device

Dear Sir:

This is the seventh of a series of progress reports addressing the status of the referenced contract. This report covers the period January 21 to February 28, 1982.

STATUS

Modification number one to the reference contract was received at Georgia Tech on 19 February. The Mod changed Task B and added $8,138 to the total contract price. As pointed out in the January progress report under "PLAN", a delay of 3 weeks in project development has resulted. All required CP/M equipment and software has veen ordered, received and installed.

PLAN

Effort will be devoted to adapting the MMI software code to the CP/M operating system. A demonstration of MMI capability is tentatively set for the week of 12 April 1982.

PROBLEM AREAS

None

FUNDING STATUS

Initial Contract Budget	$80,039.00
Plus addition under Mod I	8,138.00
New Full Contract Budget	88,177.00
less dollars expended	84,237.70
Balance as of 3/31/82	$ 3,939.30

Sincerely,

Billy B. Wise
Project Director
Command and Control Programs
Electronics & Computer Systems
Laboratory

Approved:

H. Bennett Teates
Head, Command and Control Programs
Electronics & Computer Systems Laboratory

BBW/HBT:nr

A-3008

Georgia Institute of Technology

ENGINEERING EXPERIMENT STATION

ATLANTA, GEORGIA 30332

May 10, 1982

OICC
Naval Civil Engineering Center
Port Huenume, California 93043

SUBJECT: Deliverable Number 10, April Letter Progress Report (A-3008)

REFERENCE: Contract No. F33657-80-G-007 ZZ01, Man-Machine Interface Device

Dear Sir:

This will be the final monthly progress report addressing the status of the referenced contract. This report covers the period April 1 to April 30, 1982.

Status

The problem with trapping touch panel interrupts has been solved. After having been granted access to Chromatics' proprietary schematics, it was determined by Georgia Tech engineers that the documentation on the CG-3999 series computer was in error. With this knowledge, the elusive solution to the interrupt problem was readily discovered and implemented. The result is that the MMI demonstration device is working in accordance with the original design plan. Device operating speed/response time is very representative of that of an actual EMCS, and it gives a plausable demonstration of the touch interactive color graphics technique for EMCS control. On Tuesday, April 13, 1982, the MMI device was demonstrated to the contract technical representative and a group of Tri-service EMCS representatives. Suggestions for modifications to the device software were received and implemented as appropriate.

Plan

Final corrections to the documentation will be completed and deliverables will be provided as scheduled. It is planned to deliver the following in fulfillment of contract obligations as understood on this date:

1. Man-Machine Interface Device Software Design Manual (includes software listings).

2. Man-Machine Interface Device Operators Manual.

3. Man-Machine Interface Device Operating Software program (on 8" floppy diskette).

The man-machine device hardware and documentation will remain at Georgia Tech as GFE for use in follow-on project work to implement on EMCS operator training device.

Problem Areas

None

Funding Status

Initial Contract Budget	$80,039.00
Plus addition under Mod I	8,138.00
New Full Contract Budget	$88,177.00
Less dollars expended	87,870.72
Balance as of 4/30/82	$ 306.28

Sincerely,

Billy B. Wise,
Project Director
Command and Control Programs
Electronics & Computer Systems
Laboratory

Approved:

H. Bennett Teates
Head, Command and Control Programs
Electronics & Computuer Systems Laboratory

BBW/HBT:dar

GEORGIA INSTITUTE OF TECHNOLOGY
Engineering Experiment Station
Atlanta, Georgia 30332

Man-Machine Interface Device

Interim Software Design

Design Technical Report

by

B. S. Rice

R. C. Coleburn

B. B. Wise

of the

Electronics and Computer Systems Laboratory

GIT/EES Project A-3008

September 1981

Prepared for

NAVY CIVIL ENGINEERING LABORATORY
Port Hueneme, California

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

MAN-MACHINE INTERFACE DESIGN SPECIFICATIONS

1.0 INTRODUCTION

1.1 Objective

The objective of this project is to develop a prototype man-machine interface (MMI) device which will demonstrate the concept of interactive color graphics as an operator control medium for energy monitoring and control systems (EMCS). The development of and supporting rationale for the interactive color graphics concept was described in detail in a report entitled "A Man-Machine Interface for Energy Monitoring and Control Systems," dated March 1981 (CEL CR 81.013). This concept is intended to provide a "friendly" computer interface which will allow unsophisticated users to efficiently interact with a complicated EMCS, with a minimum amount of training required. The purpose of this report is to document the MMI device operating capabilities and software program design.

1.2 Basis of Design

The MMI device is intended to demonstrate the technique of interactive color graphics control of an EMCS. It is in no way intended or construed to simulate the functioning of an actual EMCS. The basis for the operating capabilities of the MMI device is contained in two documents, (1) "Guide Specification for Large Energy Monitoring and Control Systems" (CEGS 13947), and (2) "Design Manual for Energy Monitoring and Control Systems" (NAVFAC DM 4.9). The guide specification lists twelve tasks to be accommodated by an EMCS operator terminal, including:

(1) Request a display of any digital or analog point, or any group of related points in the system.

(2) Startup and shutdown selected systems or devices.

(3) Initiate reports.

(4) Request graphic displays.

(5) Modify time and event scheduling.

(6) Modify analog limits.

(7) Adjust setpoints of selected controllers.

(8) Select manual or automatic control modes.

(9) Enable and disable individual points; disabling shall take precedence over all other actions.

(10) Enable and disable individual FIDs.

(11) Enable and disable individual MUX or IMUX panels.

(12) Point definition.

Tasks (1) through (11) were considered to be within the scope of the lowest level operator's attention, and formed the basis for the operating capabilities of the MMI device. Task (12) was felt to be the province of a higher level operator, and was not implemented directly in the MMI device.

The approach taken was to develop HVAC subsystem color graphic diagrams, similar to those contained in NAVFAC DM 4.9, and make them interactive through use of an infrared touch panel integrated with the CRT display. This touch panel allows the operator to send commands to the EMCS by simply touching the CRT screen with a finger, reducing to a minimum the requirement to type on the keyboard and eliminating confusing computer language-oriented input commands. A subgoal was to make the EMCS itself virtually transparent, so that the operator would have the feeling of interacting directly with the HVAC system, rather than with the EMCS computer. Figures 1, 2 and 3 are examples of the HVAC graphics diagrams which are to be implemented in the MMI device.

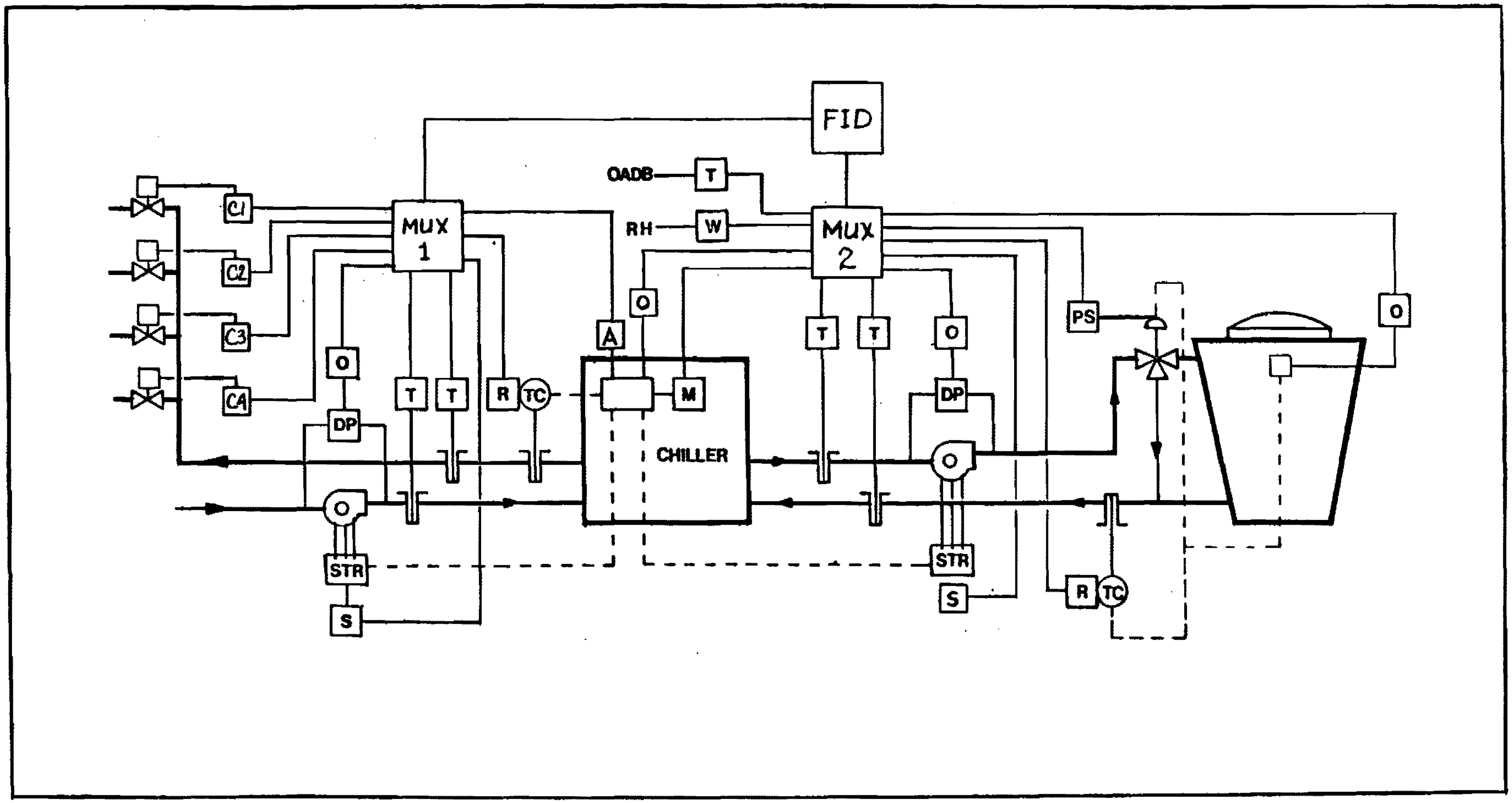


Figure 1. Water Cooled Chiller

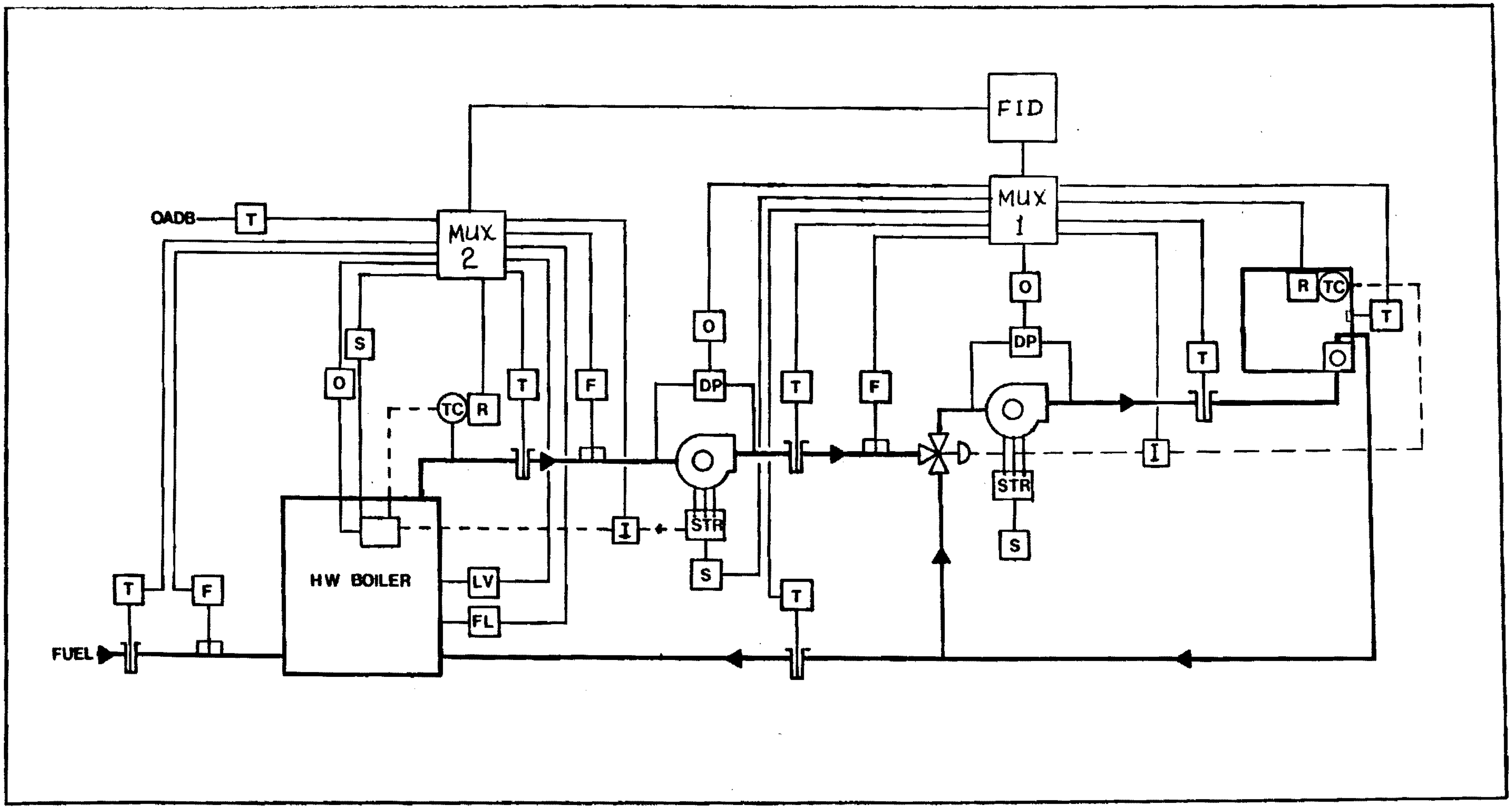


Figure 2. Hot Water Boiler and Radiation System

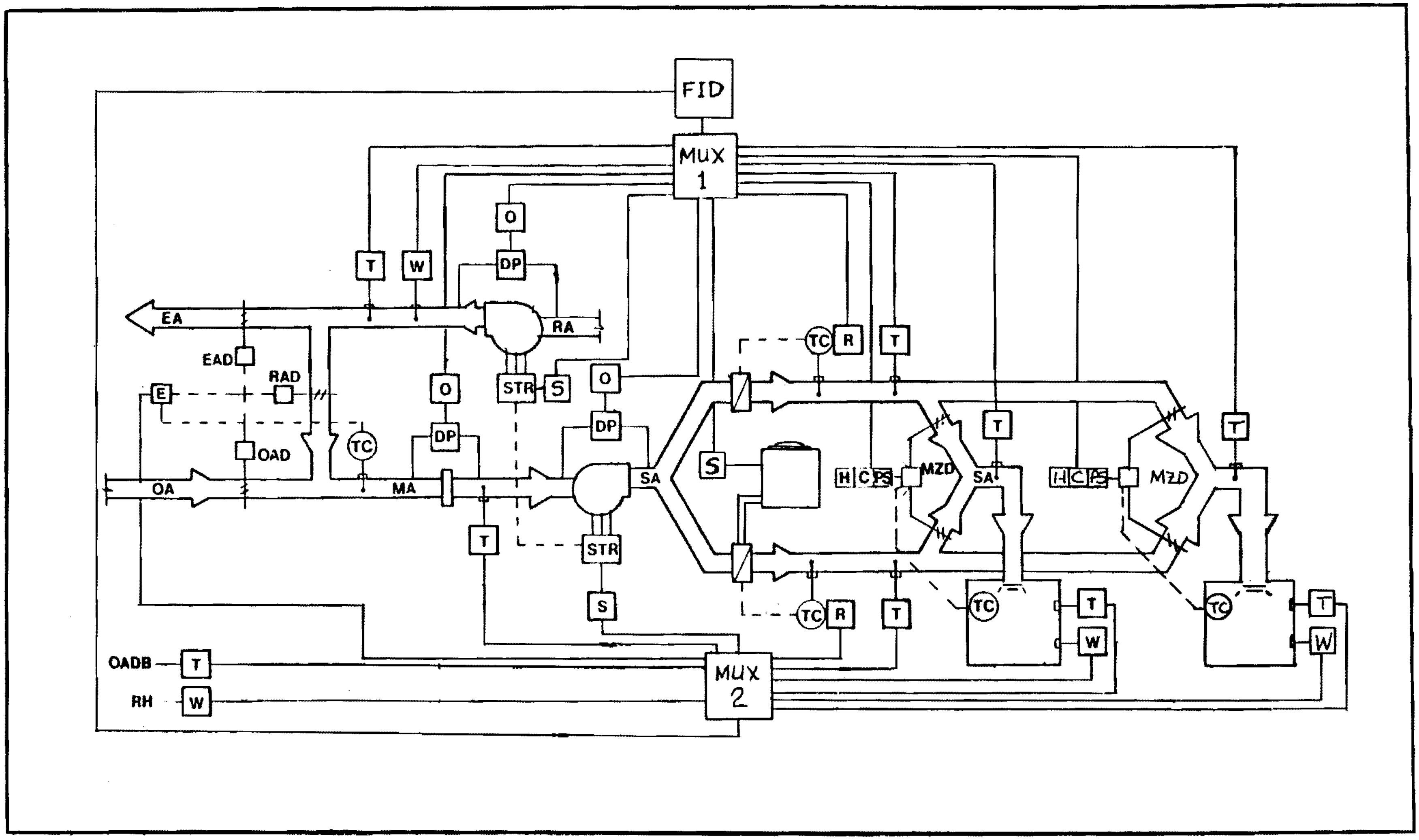


Figure 3. Multizone DX-AC Unit

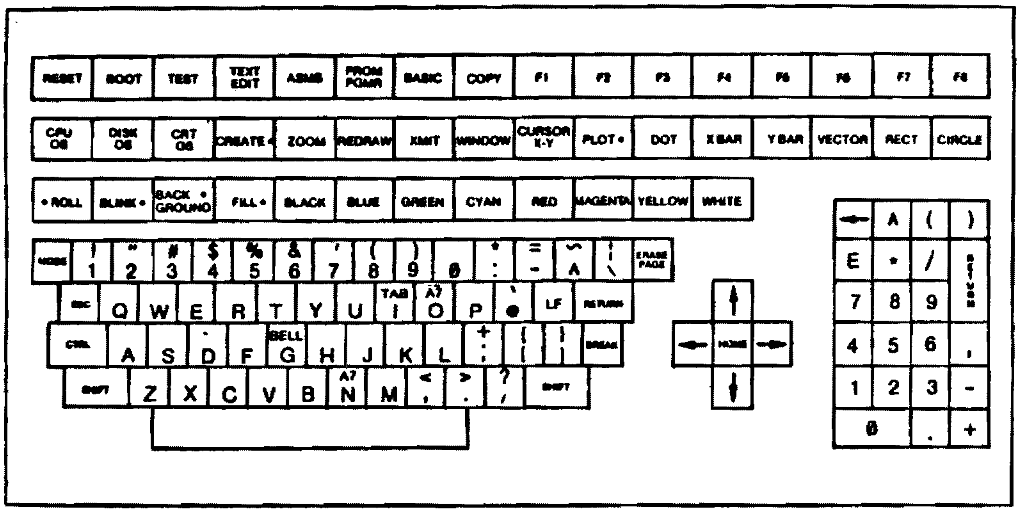
1.3 Hardware Description

As documented in CEL CR 81.013, there are several hardware features which must be present to enable satisfactory operation of an operator process-control interface. Among them are rapid-draw and polygon fill, to minimize graphics drawing time; high resolution, large size CRT display, to enable accurate and uncluttered reproduction of HVAC system graphic diagrams; fast CRT refresh rate, to minimize operator eyestrain and fatigue; and a positive reliable interactive touch capability, to minimize errors. It was specified that the MMI demonstration device be stand-alone in operation (i.e., that it require no host computer to perform its demonstration function). This stand-alone feature was specified to facilitate moving the MMI device between locations for conducting demonstrations.

Based upon extensive past experience with color graphics equipment, and a survey of the latest equipment readily available in the market place, the color graphics computer line produced by Chromatics, Inc. was selected for use in the MMI device implementation. The specific model chosen was the Chromatics CG3999, featuring a 19-inch, high resolution (512 x 512 pixel matrix) CRT with a 60 Hz non-interlace refresh rate. A number of available options were selected to obtain the required features enumerated above, such as complex boundary fill, extended graphics, blink, and a function key processor. In addition to these hardware color graphics capabilities, the CG3999 can operate as a stand-alone computer with its Z-80 microprocessor, 32K RAM and dual eight inch floppy disc system. To accommodate the touch panel, and to allow for system growth, three extra RS-232 interface ports were provided. While not utilized in the proposed MMI concept, a light pen was acquired to permit flexibility in color graphics system design. The keyboard accompanying the CG3999 has 128 keys (Figure 4), including numerous special function keys to activate built-in color graphics generation capabilities. This keyboard will be used to generate HVAC system graphic diagrams during the development process, and could be used by installation system programmers in the field. However, in keeping with the goal of interface simplicity and transparency for the lowest level operator, a more less complex detachable typewriter-like keyboard, with a numeric key pad and minimal extra keys, will be provided for use with the MMI device (Figure 5).

Of the several touch feature technologies available in the industry, the beam interrupt type was chosen as being most durable, reliable and error-free. The IR beam interrupt touch panel from Carroll, Inc. comes in kit form and includes a printed circuit "window frame" with IR LED emitters and phototransistor diodes and its own scanning and decode logic circuits. The "window frame" is mounted around the face of the color graphics CRT and provides a grid of IR beams across the face of the tube. Resolution as small as 1/8 inch can be obtained with this device. As built-in scanning circuits activate the LED's in sequence, the circuitry detects a broken IR beam resulting from an operator touch on the CRT face, and the decode logic determines the X-Y position of the touch or hit. The touch panel decode circuit connects to the Chromatics set via an RS-232 interface port at the rear of the CRT cabinet. The touch panel transmits a sequence of 3 or 4 ASCII characters each time the touch system is activated. The first character is a unique uncover code that identifies the following two characters as touch data. The second and third characters represent the coordinates of the broken X and Y beams. The fourth character, which is optional, is a stop code.

To allow for hard copy output, and to demonstrate report printing capabilities, a printer has been provided. As specified, the printer provided is a NEC Model 7715 Spinwriter. This printer offers letter quality text and high resolution plotting/graphing output at 55 CPS max print rate. It attaches to the Chromatics via an RS-232 interface port. Also provided with the NEC printer is a cut sheet feeder and a vertical forms tractor. Appendix A contains a complete listing of MMI device equipment.



NOTE: • INDICATES ILLUMINATED KEY.

Figure 4. 128-Key Keyboard

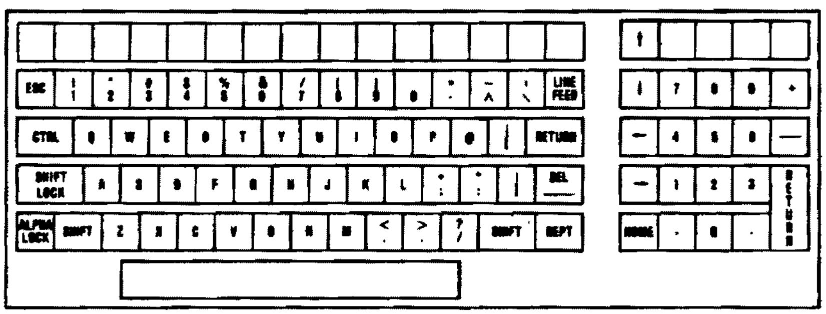


Figure 5. Simple Keyboard

2.0 OPERATIONAL DESCRIPTION

In all phases of this design, special consideration has been given to the human factors associated with effectively utilizing an EMCS system. These factors are discussed in the following paragraphs, along with justification for the choice made between alternatives.

2.1 Display Screen

The screen is a CRT display unit with color graphics capability. The screen has a 512 x 512 visible dot resolution. Eight screen colors are available: red, green, blue, magneta, cyan, yellow, white, and black. The screen is divided by the software into five major areas (see Figure 6):

(1) Date/Time/Operator Continuous Display

(2) Graphics Display Area

(3) Special Function Keys

(4) Message/Text Area

(5) Alarm Indicator

By assigning the types of displayed information to a consistent location on the screen, the operator will know where to look for specific information without having to scan the entire screen. Further, vital information (such as the alarm indicator) will always be present.

The CRT display unit is also equipped with an interactive touch panel utilizing infrared beam interruption technology. The touch panel allows the operator to directly interact with the graphics display, thus minimizing the use of a keyboard.

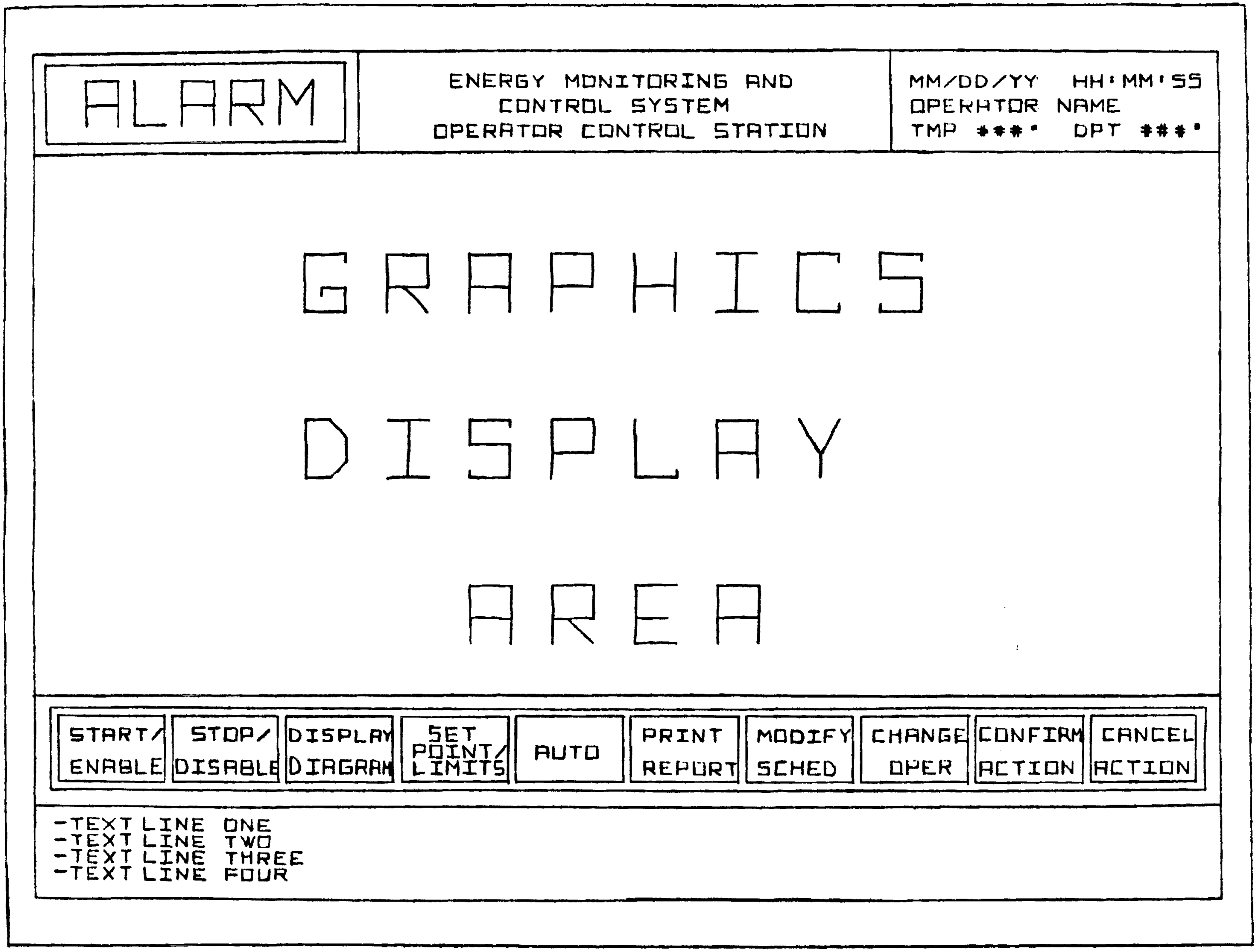


Figure 6. Screen Layout

2.2 Date/Time/Operator Continuous Display

The Data/Time/Operator Continuous Display is located in the upper right corner of the screen. This area's background color is yellow, with black lettering used to display the current operator's name, the date, time, outside temperature, and dewpoint. Colors were chosen for optimal legibility.

2.3 Graphics Display Area

The Graphics Display Area is used for multi-color graphic display of data environment diagrams, as well as for menu selection. It occupies the middle two-thirds of the screen. Its background color is black, to provide the best background for the graphics by increasing contrast and reducing "noise." Each data environment in the system has been diagrammed using HVAC symbols and the graphics capability of the computer. The operator can command the system to display any of these diagrams for inspection or use in modifying the operation of its associated data environment. Figures 1, 2 and 3 are illustrations of sample data environment (DE) diagrams.

2.4 Special Function Keys

There are ten touch-activated Special Function Keys aligned in a row slightly below the horizontal center of the screen. These keys are the primary means by which the operator interacts with the EMCS. The keys are arranged in an order which places the most critical and frequently used functions on the ends, thus reducing operator search time. In addition, related functions are placed together, and there is a left-to-right ordering of complementary keys representing positive-to-negative connotations (e.g., START precedes STOP). In this way an operator will have less difficulty in finding keys, thus improving response time in critical situations. The touch key area is set off from the rest of the screen by a rectangular background border of cyan. The cyan background is bordered by blue lines on both the inner and outer edges. The keys themselves are located on a panel strip of black inside this cyan background border. Each key is color coded (explained below).

2.5 Message/Text Area

The Message/Text Area is located across the bottom of the screen. Its background color is black and all messages appear in a color chosen for good visibility and appropriate connotations. This area is used by the MMI to provide information to the operator or prompts for a response. Most of the cues which are written to this window are described in Section 2.8 through 2.8.10. This area is also used to display error messages and alarm conditions. A complete list of all cues, error messages, alarm messages, and their descriptions is given in Appendix B. The color code scheme for messages can be found in Table 1.

2.5.1 Visual Message Indicators

Throughout this manual, a change in color (always to yellow) indicates that a selection has been made. Use of flashing displays or text is strictly reserved for alarm conditions. Thus, whenever an operator sees something flashing on the screen, he can always interpret it as some type of alarm message or indicator. Likewise, whenever an item changes color to yellow, it can always be interpreted as a message indicating selection. Messages which appear in the text area are also color coded. Cues appear in green because green connotes "action" and a cue is a prompt signaling the operator to take action. Error messages appear in red, since red can be used to connote an abnormal condition. Alarm messages appear as white text on a flashing red background, accompanied by an audible tone. Thus, the color for alarm messages matches the alarm indicator. See Table 1 for a complete color code scheme and Table 2 for common color connotations.

MESSAGE COLORATION SCHEME

Message Type	Color Representation
Cue (Prompt)	Green lettering on black background
Error Message	Red lettering on black backgroung
Alarm Message or Cue	White lettering on flashing red background

TABLE 1

COMMON COLOR CONNOTATIONS

Reds:	Stop, Abnormal Condition
Greens:	Start, Go, Action, Ready
Blues:	Calm, Normal

TABLE 2

2.6 Alarm Indicator

The Alarm Indicator is located in the upper left corner of the screen opposite the Date/Time/Operator window. It appears as a flashing red rectangle with the word "ALARM" printed in white. The red rectangle is bordered by a non-flashing white border. The flashing red indicator is displayed only when an alarm condition exists; however, the white border will always be present on the screen. In addition to this visual indicator, an audible tone will sound when an alarm condition is detected.

2.7 Operation of Special Function Keys

The operation of each function key follows the same general pattern, except that steps 1 and 3 may be reversed, since they are order-independent:

(1) A function key is touched and becomes backlighted in yellow, indicating that it has been selected. Backlighting is defined as changing the background color of a key or symbol on the CRT screen. Only one function can be selected at a time. If another key is touched (as in step 5b), the first choice will be cancelled. Thus, the yellow backlight will always indicate which function key has been selected.

(2) A cue appears in the text area instructing the operator to touch the desired device symbol or menu item (or function key if steps 1 and 3 are reversed).

(3) The operator touches the desired device symbol or menu item as instructed by the cue, and the border of the device symbol (or the entire menu selection) changes color to yellow to indicate that it has been selected. This color change is referred to as highlighting. Only one device or menu item can be selected at one time, so only one device symbol or menu item can be highlighted with yellow at one time. Thus, the yellow highlighting will always indicate which device or item has been selected.

(4) A cue appears instructing the operator to touch the CONFIRM ACTION key to execute the selected command.

(5) The operator now has three choices:

a. Touch the CONFIRM ACTION key to execute the command.

b. Select a new function, device, or menu item, i.e., go back to step 1 or 3.

c. Touch the CANCEL ACTION key to abort the current command and return the screen to the way it was before any keys were touched.

2.7.1 Design Modifications

In the original design specifications (CR 81.013), several keys were designed as split-function keys. Since some people have large fingers, a split-function key could pose problems in that the touch target area could be small enough to cause difficulty in selecting the desired function. Therefore, the original design has been modified to eliminate the need for split function keys. Several other modifications have been made with regard to the set of keys. Discussion of these additional modifications and justification for them is interspersed in the function key descriptions below. Keys are discussed in order from left to right on the screen.

2.7.2 Keys That Are Environment Dependent

Note that the START/ENABLE, STOP/DISABLE, SET POINT/LIMITS, AUTO/MANUAL, and MODIFY SCHED keys are only valid if a data environment diagram is currently being displayed. If one of these keys is touched and this is not the case, then the error message "-You Must Use DISPLAY DIAGRAM To Get A Diagram Before You Can Use This Command" appears in the text area, and then after a pause the screen appears the same as it was before the key was touched.

2.7.3 Symbol Coloration

The coloration of a device symbol will always reflect its current status, according to the following conventions:

Blue Border	-- Non-selected symbol
Yellow Border	-- Selected symbol (only one at a time)
Green Interior	-- Device in operation (on) under automatic control
Black Interior	-- Device not in operation (off) under automatic control
Diagonally Striped Interior	-- Device in manual mode (For a device which has been turned off manually, the striping will be black and white. For a device which has been turned on manually, the striping will be green and blue).

If the DE is operating in automatic mode, all devices whose interior symbol color is green have been started/enabled by the automatic control. If a symbol's interior color is diagonally green and blue striped, then that device has been manually started/enabled and the automatic control for that device has been overridden. Diagonal black and white striping is used to indicate that the device has been manually stopped/disabled and that the automatic control for that device has been overridden. A symbol whose interior color is black has been stopped/disabled by the automatic control, (see Table 3). Diagonal striping is used to indicate a manual override for the following reasons:

(1) A manual override is an exceptional case, and the operator needs to be visually reminded that he has overridden the automatic control. This visual indicator should be easily distinguishable from the rest of the display.

(2) Diagonal lines will stand out against the many vertical and horizontal lines on the diagram being displayed.

(3) Striping is not as uncomfortable or fatiguing to the eyes as is checkerboarding, or other patterns. This is especially true for operators who have astigmatism.

SYMBOL COLORATION SCHEME

DEVICE STATUS	DE's MODE OF OPERATION: AUTO	MANUAL
STOPPED/ DISABLED	BLACK INTERIOR	BLACK/WHITE STRIPED INTERIOR
STARTED ENABLED	GREEN INTERIOR	GREEN/BLUE STRIPED INTERIOR
SELECTION	YELLOW BORDER	YELLOW BORDER
ALARM	FLASHING SYMBOL	FLASHING SYMBOL

TABLE 3

2.8 Function Key Descriptions

In the function key descriptions which follow, it is assumed that the reader understands the general pattern of operation outlined above. Therefore, certain repetitious details will be omitted. Deviations from the pattern will of course be included. The description for the START/ENABLE key will include more detail than that of the remaining nine keys so as to allow the reader to become more familiar with the pattern.

2.8.1 START/ENABLE Key

The START/ENABLE key is used to manually start mechanical devices and enable monitoring and control devices. The word "device" is used here to refer to any entity in the system which can be operated independently, whether it be a FID, MUX, IMUX, pump, fan, or data point (analog or digital input or output). In the original design, the start and stop functions were on a single key, as were the enable and disable functions. However, it was felt that placing start and enable on the same key and, conversely, stop and disable on another key, would provide an easier means of control for the operator, for the following reasons:

(1) Start and enable have the same connotation, differing only in that they are applied to different types of physical entities, and likewise for stop and disable;

(2) Start and enable are mutually exclusive functions for stop and disable;

(3) The software is capable of distinguishing which devices are stopped/started and which are enabled/disabled.

In this way, the operator need not remember which devices must be started/stopped as opposed to those which must be enabled/disabled. This design also eliminates the need for split keys, as was discussed in section 2.7.1. In addition, the START/ENABLE key is colored green, which connotes "go" or "start," to further aid the operator in rapidly associating its color with its function and in scanning for the key.

The START/ENABLE key is used in the following manner. When a data environment (DE) is displayed on the screen, the operator can start or enable a device by touching the START/ENABLE key and the appropriate device symbol on the graphic display (order is independent). When the key is touched, it changes color to yellow (hereafter referred to as backlighting the key), indicating that it has been selected. Likewise, when a device is selected, its border changes color to yellow (hereafter referred to as highlighting), indicating that it has been selected. If the key is touched first, the cue "-Touch Appropriate Device Symbol" appears in text area. If the device symbol is touched first, then the cue "-Touch Desired Function" appears in the text area. Once both a function and a device have been selected, the cue "-Touch CONFIRM ACTION To Execute" appears in the text area. At this point the operator can do one of three things:

1. Touch the CONFIRM ACTION key, at which time it becomes backlighted; the command to start/enable the device is sent to the EMCS; the START/ENABLE key backlight is extinguished; the border of the device symbol reverts back to its original color, and its interior changes to an appropriate color (see Section 2.7.3) to indicate that it has been started/enabled; the message "-Device (x) Has Been (Started/Enabled)" appears in the text area; and finally, after a slight pause (representing the time needed for the requested action to occur), the CONFIRM ACTION key backlight is extinguished and the text area is erased.

2. Cancel the entire operation by touching the CANCEL ACTION key, at which time it becomes backlighted; the START/ENABLE key's backlight is simulaneously extinguished; the device symbol border color returns to its original color; the message "-Command Action Cancelled" appears in the text area, and finally the text area is cleared. The screen now appears the same as it did before the operator touched any keys or symbols.

3. Touch a different device symbol and/or function key, at which time the appropriate symbol border color changes are made and/or the function key backlights are changed, and appropriate cues are given.

2.8.2 STOP/DISABLE Key

The STOP/DISABLE key is used to manually stop mechanical devices and disable monitoring and control devices. It operates in the same manner as the START/ENABLE key. This key is colored red because red connotes "stop", and this further aids the operator in distinguishing its function. See Table 3 for an illustration of appropriate device symbol interior colors, based on the current operating mode.

2.8.3 DISPLAY DIAGRAM Key

The DISPLAY DIAGRAM key is used to display HVAC diagrams of specific data environments (DE's) in the graphics display area. It is colored blue, which connotes "calmness," since its use does not directly alter the operation of the system. When the key is touched, it becomes backlighted and the building or floor selection menu is presented in the graphic area (see Figure 7). The cue "-Touch Square Beside Building (Floor Plan) Desired" appears in the text area. The menu is presented in cyan and the squares are color-filled targets. When touched, the square and its text description change color to yellow to indicate which data environment has been selected, and the cue "-Touch CONFIRM ACTION To Execute" appears in the text area. If the operator touches the CONFIRM ACTION key, it becomes backlighted, the DISPLAY DIAGRAM key backlight is extinguished, the selected diagram appears in the graphics area, and finally the CONFIRM ACTION key backlight is extinguished. Alternately, the operator can change the menu or function selections, or touch the CANCEL ACTION key to cancel the command.

2.8.4 SET POINT/LIMITS Key

The SET POINT/LIMITS key is used to adjust set points and analog limits for appropriate HVAC devices. This key is magenta in color. A DE diagram must be present on the screen when this key is touched; otherwise, the MMI will respond in a manner similiar to that described

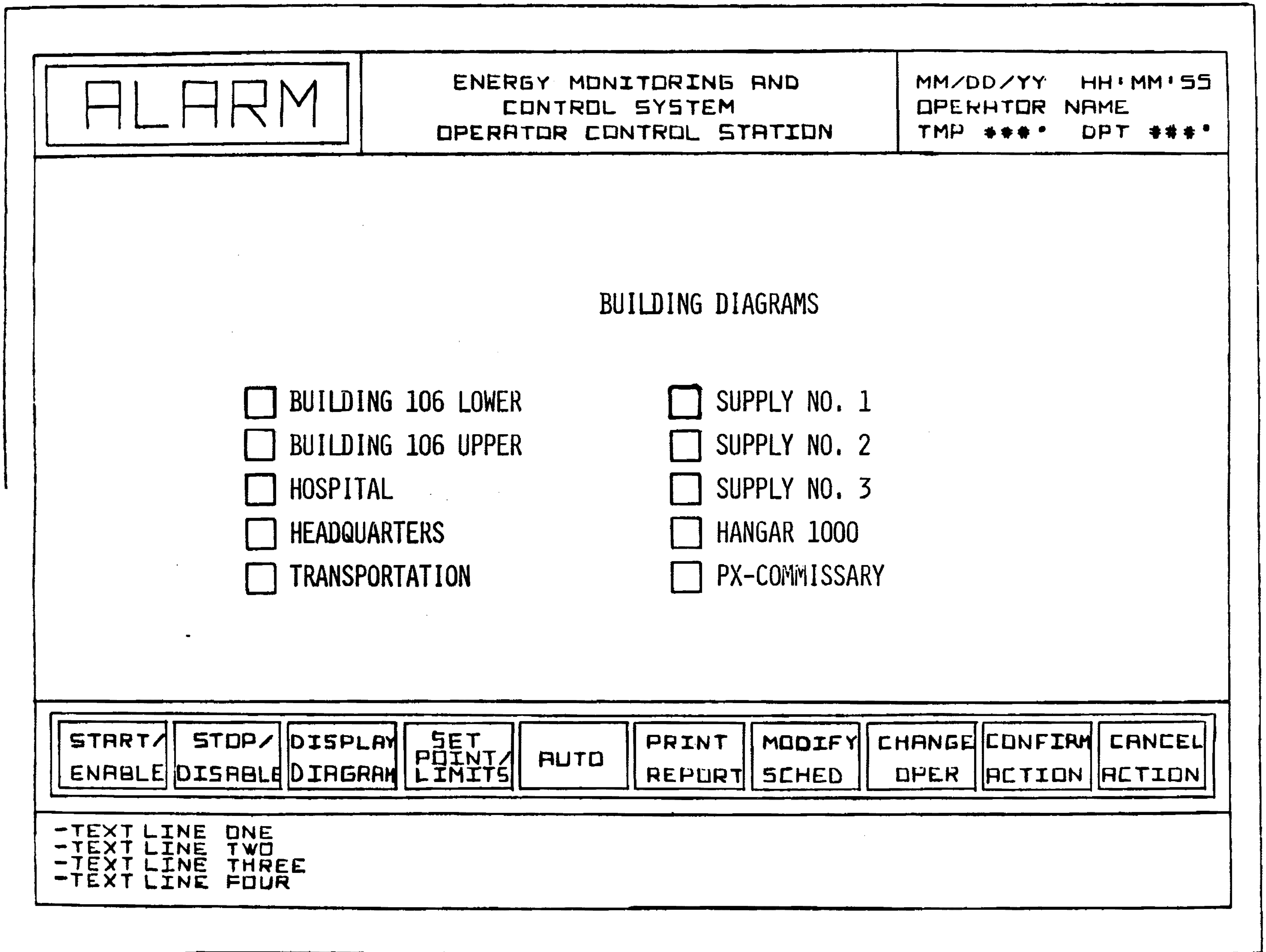

ALARM
ENERGY MONITORING AND
CONTROL SYSTEM
OPERATOR CONTROL STATION
MM/DD/YY HH:MM:SS
OPERATOR NAME
TMP ***° DPT ***°
BUILDING DIAGRAMS
BUILDING 106 LOWER
BUILDING 106 UPPER
HOSPITAL
HEADQUARTERS
TRANSPORTATION
SUPPLY NO. 1
SUPPLY NO. 2
SUPPLY NO. 3
HANGAR 1000
PX-COMMISSARY
START/ ENABLE
STOP/ DISABLE
DISPLAY DIAGRAM
SET POINT/ LIMITS
AUTO
PRINT REPORT
MODIFY SCHED
CHANGE OPER
CONFIRM ACTION
CANCEL ACTION
-TEXT LINE ONE
-TEXT LINE TWO
-TEXT LINE THREE
-TEXT LINE FOUR

Figure 7

in Section 2.7.2. When this key is touched it becomes backlighted, and the cue "-Touch Appropriate Device Symbol" appears in the text area. If the symbol was touched prior to touching the key, the cue "-Touch Desired Function" would have appeared in the text area. If the function is not appropriate for the device, the message "-You Cannot Set Points/Limits For That Device -- Try Another" appears in the text area. When both function and device have been selected, the cue "-Touch CONFIRM ACTION to execute" appears in the text area. At this point the operator can change the function selection, or use the CANCEL ACTION key. If the operator touches CONFIRM ACTION, a menu of choices is presented in cyan: one for the set point, one for the low limit, and another for the high limit. Each choice will have a touch-sensitive square beside it. To modify any or all points/limits, the operator merely touches the desired selections, one at a time, and the cue "-Please Type In The New Value" appears in the text area. The choice selected will always change color to yellow. When all of the points/limits appear as desired, the operator merely touches CONFIRM ACTION, at which time the graphics area returns to the previous diagram, the key backlights go out, and the message "-Command Action Completed" appears in the text area. Note that the operator could still have decided not to make any changes simply by using the CANCEL ACTION key.

2.8.5 AUTO/MANUAL Key

The AUTO/MANUAL function key is a special toggle key which serves three purposes. It functions as a status indicator as to which mode the particular data environment (DE) is operating under, as well as a means of selecting the operating mode for the DE or an individual device. It is colored white so that it is easily distinguished from the other keys (no other key is white) and placed in the middle of the screen so that it can be easily seen. The key has three different representations on the screen:

(1) When no DE is being displayed, the key will have the words "AUTO/MANUAL" printed on it in black letters to indicate its function.

(2) When a DE is displayed which is running in automatic mode, the key will have the word "AUTO" printed on it in blue letters indicating that the DE's mode of operation is automatic. Blue lettering is used since blue connotes "calmness," and automatic mode is "calm" since it requires no intervention from the operator under normal circumstances.

(3) When a DE is displayed which is running in manual mode, the key will have the word "MANUAL" printed on it in green letters indicating that the DE's mode of operation is manual. Green lettering is used because green connotes a feeling of "movement" or "action" (e.g., go, start, ready), and when in manual mode the operator must take direct action to control the DE.

Touching the AUTO/MANUAL key when a DE is not displayed will generate the response indicated in Section 2.7.2, after which the screen returns to its previous state. When a DE is displayed and the key is touched and no device has been selected, the key will backlight, and the cues "-Manual Mode Selected For DE" (or "-Auto Mode Selected For DE") and "-Touch CONFIRM ACTION To Execute" will appear in the text area. If CONFIRM ACTION is touched, the message "-Command Action Completed" will appear in the text area, the function key will toggle to indicate the new mode of operation, the backlight will extinguish, and the screen will reflect the current status of the DE. If a device has been selected when the key is touched, the key will backlight, and the cues "-Manual Mode Selected For Point" (or "-Auto Mode Selected For Point") and "-Touch CONFIRM ACTION To Execute" will appear in the text area. If CONFIRM ACTION is touched, the message "-Command Action Completed" will appear in the text area, the key and symbol backlights will extinguish, and the symbol will become diagonally striped to reflect its new mode of operation. Otherwise, the operator can change his mind by selecting a different function or device or by using CANCEL ACTION. Note that if the function key is touched first, the operator can still select a device if he so chooses. Appropriate cues will always be given to inform the operator of whether the mode is being changed for the DE or a device.

2.8.6 PRINT REPORT Key

The PRINT REPORT key is used to allow the operator to initiate printing of special reports. It is colored blue because blue connotes "calmness," and use of this key does not disturb the system's operation. Touching this key causes it to be backlighted, and a menu of available reports colored in cyan to appear on the screen. The message "-Touch Square Beside Desired Report" appears in the text area. Touching a square elicits the cue "-Touch CONFIRM ACTION To Execute," and the choice is highlighted in yellow. If CONFIRM ACTION is touched, the message "-Command Action Completed" is written in the text area, after which the key backlights go out and the screen returns to its previous status. The requested report is then sent to the printer. Alternately, the operator may select a different report or function, at which time the screen is updated to reflect the change, or use CANCEL ACTION to inhibit any report. See Figures 8, 9, 10, and 11 for sample report formats.

2.8.7 MODIFY SCHED Key

The MODIFY SCHED key is used to modify the automatic schedule of operation for a particular data environment (DE). Like the SET POINT/LIMITS key, it is colored magenta to indicate that these functions are related in that they both modify stored operating parameters. It must be used in conjunction with a DE currently displayed in the graphics area (see Section 2.7.2). Once touched, it becomes backlighted, and a menu of schedule selections for the DE appears on the screen. This schedule can be modified in the same manner as that described for setting points/limits. CANCEL ACTION and CONFIRM ACTION also operate as described previously.

FRIDAY 07/31/81 10:00:32

ENERGY UTILIZATION SUMMARY

Report on: (Point, Unit, Area, Building, EMCS, etc.)

Total Energy Usage		Total Energy Usage		Max Rate Consump		Max Rate Consump		Outside	Air
								Temp	Hum
07/30/81	07/31/81	06/81	07/81	07/30/81	07/31/81	06/81	07/81		
#BTU	#BTU	#BTU	#BTU	#BTU	#BTU	#BTU	#BTU	°F	%RH

FIGURE 8

ANALOG LIMIT SUMMARY

Report as of: MM/DD/YY HH:MM:SS

Report on: (Point, Building, all EMCS, etc.)

POINT ID	VALUE	LIMITS LOW	LIMITS HIGH	DIFF
Mess Hall, TC2	75°F	72°F	78°F	6°F

FIGURE 9

LOCK OUT SUMMARY

MONDAY	MM/DD/YY	HH:MM:SS

Points Disabled:

West Bldg, Floor 2 -- Fan 3 disabled

Rec Bldg, Floor 1 -- Fan 1 disabled
Rec Bldg, Floor 1 -- Smoke Alarm 3 disabled

Mess Hall -- Outdoor Damper 4 disabled
Mess Hall -- Temperature Controler 2 disabled

FIGURE 10

ALARM SUMMARY

Outstanding Alarms as of: MM/DD/YY HH:MM:SS

ALARM POINT IDENTIFICATION	TIME OCCURED	CLASS	LIMITS LOW	LIMITS HIGH	CURRENT VALUE/STATUS
Mess Hall, TC2	12:45:32	2	72°F	78°F	79°F
Rec Bldg 3, Fan 1	14:05:10	3	----	----	Unit Failure

FIGURE 11

2.8.8 CHANGE OPER Key

The CHANGE OPER Key is a multipurpose function key. It stands for "Change Operator/Operation." It allows the operator to shutdown the system, request HELP, or Change Operators. When this key is touched, the system enables the keyboard for input and displays the messages "-Ready To Change Operator/Operation" and "Enter Command or New Operator Name From Keyboard." There are three commands which can be entered: Operator Name, HELP, and STOP. The operator types in the desired command followed by the Return key. The operator may cancel the CHANGE OPER function by typing the Return key in response to the system request for a command, or by touching CANCEL ACTION. Operator Name refers to the actual name of the operator. If HELP or STOP is typed in, the system will behave as described in Section 2.8.8.2 or 2.8.8.3, respectively. Otherwise, the system assumes that an operator change will take place. Changing of operators is described in Section 2.8.8.1.

2.8.8.1 Operator Name Command

When an operator's name is typed in, as opposed to HELP or STOP, the system assuming that an operator change is taking place and that future commands should be recorded under a new name. It also provides system security, in that levels of operation can be defined by assigning different operators to different levels. As the operator's name is typed in, it is echoed on the screen and checked against a list of known approved operators. If the name is unknown to the system, the message "-Name Not Recognized -- Command Action Cancelled" appears in the text area and the command is not executed. Otherwise the cue "-Enter password" appears in the graphics area. As the operator's password is typed in, it is checked to see if it matches the one approved for the operator (it is not echoed on the screen). If it does not, the message "-Password Not Recognized, Access Denied" appears in the text area and the command is not executed. Otherwise, the system responds with the message "-Password Accepted. New Operator is (name)." in the text area, and the Date/Time/Operator display is updated. Use of CANCEL ACTION is permitted during this command. Use of CONFIRM ACTION is not required and will be ignored. The command can also be cancelled before the password is entered by touching a different function key.

2.8.8.2 HELP Command

The HELP command is used to enter the on-line HELP facility. HELP provides an overview of system features and capabilities. When the HELP command is entered, execution of the MMI control software will be temporarily suspended, the entire screen will be erased, and the HELP program will begin running. The operator must press the Return key to advance the screen display to the next presentation when review of the current display is complete. When all parts of the sequence have been displayed, execution of the MMI will resume from the point at which it was stopped. The operator may prematurely stop the HELP sequence at any time by typing S Return. This is the same HELP sequence which is displayed when the system is first started.

2.8.8.3 STOP Command

The STOP command is used to shut down (turn off) the MMI system. When this command is entered, the MMI begins an orderly shutdown and then displays the message "-System Shutdown" when complete. Once entered, this command cannot be cancelled.

2.8.9 CONFIRM ACTION Key

The CONFIRM ACTION key is used by the operator to signal to the system that the desired command sequence has been entered and is now ready to be executed. It is colored green because green connotes "action," and this key is used to tell the system to take action. Its use has been detailed in the descriptions of the first eight function keys.

2.8.10 CANCEL ACTION Key

The CANCEL ACTION key performs the opposite function of CONFIRM ACTION. It can be used any time prior to the actual execution of a command (i.e., before it has been confirmed) to signal to the system to abort the current command and return to the previous state. It is colored red because red connotes "stop," and this key is used to stop a current command. Its use has been detailed in the descriptions for the first eight function keys. The CONFIRM ACTION and CANCEL ACTION keys have been added to the original design for four major reasons:

(1) To reduce operator errors,

(2) To reduce the memory load on the operator,

(3) To reduce operating time, and

(4) To reduce operator frustration.

These objectives are accomplished by the CONFIRM and CANCEL ACTION keys by allowing for correction of errors and changes of mind before actual execution. Furthermore, they free the operator from the restriction of having to memorize and enter a fixed sequence of commands and parametric values.

2.9 Alarm Conditions

Alarms are usually triggered by some monitered value going out of bounds (e.g., temperature), or by an equipment malfunction. The MMI notifies the operator of an alarm and allows him to correct it in an orderly manner. The assumption behind the operational methodology described in this section is that the EMCS "stacks" alarms in a queue to be sent one at a time to the MMI on a priority assigned by the EMCS software.

2.9.1 Detection and Notification

When an alarm condition is sent by EMCS, the MMI alarm indicator will be activated and an audible tone will sound. The message "-ALARM CONDITION DETECTED" will appear flashing in red in the text area. Next, the cue "-Touch DISPLAY DIAGRAM For More Alarm Information" will appear in the text area. The audible tone will continue to sound at five-second intervals until the alarm condition is corrected, or until the operator acknowledges it. The alarm can be acknowledged by touching the alarm indicator, at which time the tone will stop sounding. This action signals to the system that the operator has acknowledged the alarm and that he will attempt corrective action. If the alarm condition still persists after two minutes have elapsed, and the operator has not attempted any corrective action during this time, the system will again begin sounding the tone at five-second intervals to remind the operator to take action.

2.9.2 Corrective Action

Following notification of the alarm, the operator can continue the current command sequence, or respond immediately to the alarm by touching the DISPLAY DIAGRAM key. Touching this key will cause the usual DE menu to appear, but with the following special indicator. The area containing the alarm will be flashing in red. Areas without alarms will appear in cyan and will not be flashing. The operator then selects in the usual manner (see Section 2.7.3) the alarm data environment to display. When the DE is displayed, the element or device in alarm condition will be flashing. The operator can then take the appropriate control action to correct the situation. In this way, the operator is in control and is not forced to respond immediately to an alarm and/or to follow a rigid correction algorithm.

When the operator makes an adjustment to the point in alarm, it is assumed that the alarm for that point will be disabled by the EMCS software for a period of fifteen minutes. (Five minutes is used in the demonstration system in order to facilitate the rapid presentation of the system's capabilities.) The fifteen minute alarm point disablement period is used to allow time for the operator's corrective action to become effective. If at the end of the fifteen minute disablement period the point is still in alarm, the operator will again be notified of the alarm in the manner described above. The process of notification, corrective action, and temporary alarm disablement will be repeated until the alarm condition has been corrected for the point. Note that the alarm is only disabled for the particular alarm point on which the operator attempted corrective action. All other points are still capable of generating alarms during the disablement period of a particular point. Note also that the operator will receive only one alarm at a time. This is based on the assumption that the EMCS computer will "stack" alarms in a queue to be sent to the MMI one at a time on a pre-determined priority basis. Therefore, in order to receive further alarms from EMCS after having received the first, the operator must acknowledge the first alarm by attempting some corrective action. EMCS will only send a new alarm when one of the following conditions exist:

(1) There are no other active alarms or,

(2) Corrective action by the operator has been attempted for all other active alarms.

During the disablement period of a particular alarm point, its device symbol will stop flashing and the disabled alarm point will appear the same as all other devices on the screen except in one respect. Its monitored value will be displayed in red, whereas the values of all other points will be displayed in cyan. In this way, the operator can visually detect any alarm points which are in the alarm disablement mode and can easily monitor their current status. By monitoring the status of an alarm point during its disablement period, the operator can determine whether or not the selected action is having a positive effect in correcting the alarm condition.

3.0 SOFTWARE DESCRIPTION

3.1 Overview

The program which will simulate the man-machine interface to an EMCS consists of an executive program and several layers of subroutines, as depicted in Figure 12. The function of the executive is to decide what is going on in the simulated real-world system (keyboard input, data base update, etc.) and to call the appropriate subroutine to respond to the system event. These subroutines in turn call other subroutines to perform specific tasks. In all, there are six levels of program modules, beginning with MMI EXEC at the highest level (Level I) and progressing downward. The levels have resulted in major part from the practice of top-down, structured program design, which stresses having a well-defined task for each module and which tends to produce easily modified and debugged software systems. Level IV (the lowest level) consists of utility subroutines which are called by modules at several different levels. Table 4 provides a brief description of the function of each of these modules, arranged alphabetically. Note that even though subroutines are referenced by line number and not by name in the BASIC language, all of the subroutines have been given mnemonic names, in order to emphasize their functions and facilitate descriptions of the system operation. For each module, a description and flowchart are provided. The page numbers in Table 4 refer to the location of the module description and flowchart in Section 3.2 This material is arranged by level, as shown on Figure 12.

Tables 5 and 6 list the elements of the two major system data bases. The MMI data base contains variables which pertain to the operation of the software system. The real-time data base contains variables whose values represent real-world system events.

Due to the limitations of the Chromatics computer and the BASIC language interpreter used in the MMI demonstration device, the keyboard and touch panel cannot be used simultaneously. Consequently, the special function keys will not be available when the keyboard has been

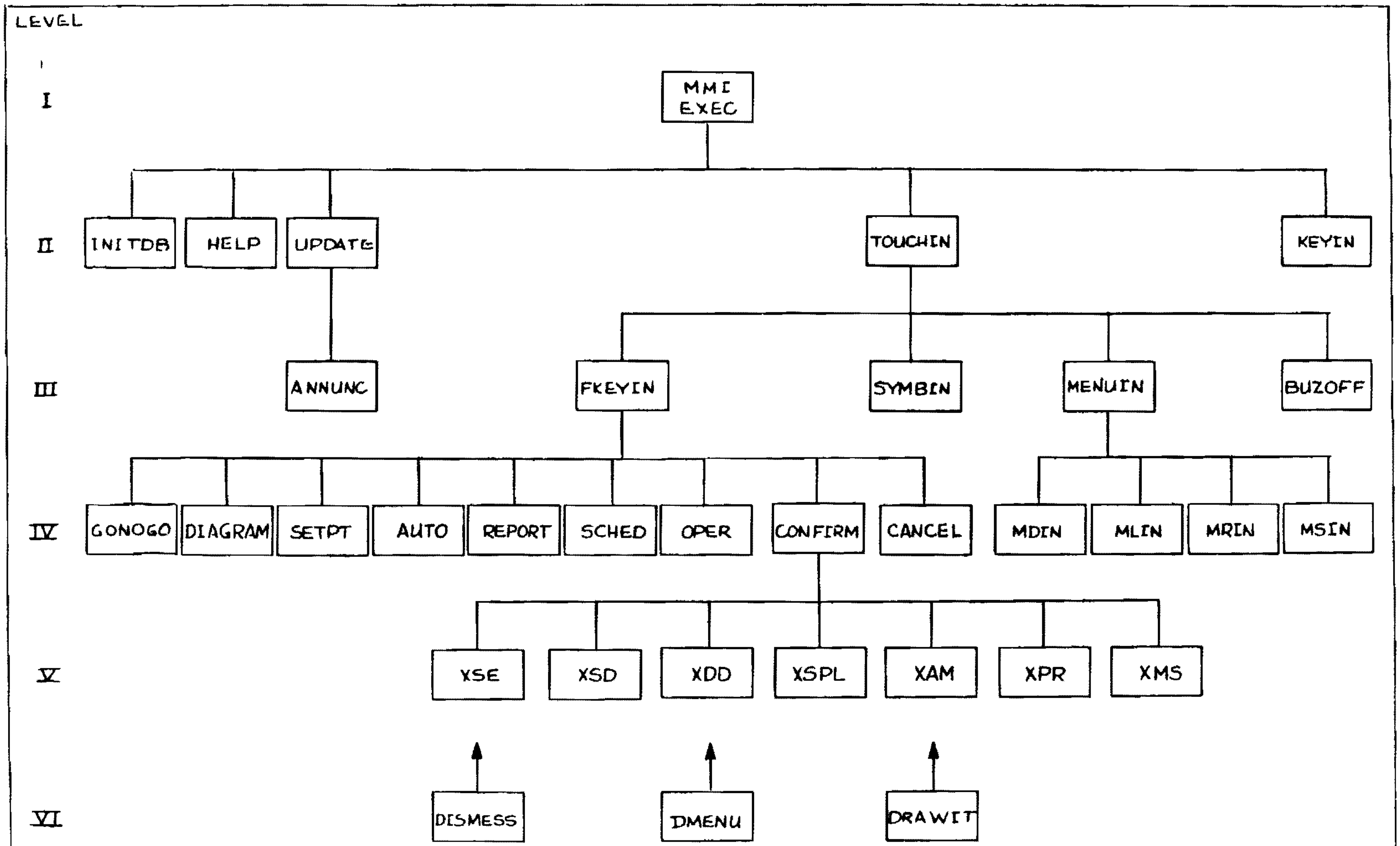


FIGURE 12. HIERARCHY OF MMI PROGRAM MODULES

enabled for input, and vice versa. This means that in the demonstration device, use of the keyboard for input will be limited to specific instances when the system requests keyboard input. The Return key is used to signal the end of a keyboard input response. Therefore, once keyboard input has been requested, the special function keys will not be available until the Return key is struck. In an actual MMI/EMCS system, these problems can be overcome through use of hardware and software with true real-time capability.

In the demonstration device, whenever a conflict arises between the selection of a special function key and a device symbol, the device symbol will always be deselected while the function key will remain selected. This feature is used to provide a standard method for resolving selection conflicts. In so doing, the assumption has been made that function selection should take precedence over device selection.

In order to facilitate the rapid presentation of the capabilities which are present in the demonstration system, time has been compressed to an approximate ratio of three minutes real time to one minute simulated time. The ratio is approximate due to the limitations of the Chromatics computer and its BASIC language interpreter. Keyboard input will always distort the time ratio. This is due to the fact that when keyboard input is enabled, all other program activity is temporarily suspended until the Return key is struck. The Return key is used to signify the end of an input response. Suspension of all other program activity is not detrimental to the demonstration device since it is a simulation.

In the demonstration device, whenever the operator disables a point, that point is assumed to be logically disconnected from the system. Complete disablement means that the device and any devices it supports are disconnected, and that values and alarm status cannot be read from that device or any other device which it supports. Likewise, control commands cannot be sent to a device which has been disabled or to a device supported by a device which has been disabled. The operator must command the system to enable a point which has been disabled before it or any devices it supports can receive control commands or send values and alarm status to the system.

TABLE 4

MODULES AND THEIR FUNCTIONS

Module	Function	Page
ANNUNC	- Annunciate change in alarm status	70
AUTO	- Process AUTO request	100
BUZOFF	- Process alarm silence request	87
CANCEL	- Process CANCEL ACTION request	120
CONFIRM	- Process CONFIRM ACTION request	115
DIAGRAM	- Process DISPLAY DIAGRAM request	94
DISMESS	- Display messages in the text area from the master message file	166
DMENU	- Place menu on screen	169
DRAWIT	- Draw or change figures on the screen (alarm indicator, highlighting, backlighting, HVAC symbols)	172
FKEYIN	- Process function key hit	74
GONOGO	- Process START/ENABLE or STOP/DISABLE request. .	90
HELP	- Provide operator instructions and assistance. .	55
INITDB	- Initialize MMI database	51
KEYIN	- Process keyboard input	67
MMI EXEC	- Man-Machine Interface executive	47
MDIN	- Process menu of diagrams	123
MENUIN	- Process menu item selection	84
MLIN	- Process menu of set points/limits	126
MRIN	- Process menu of reports	131
MSIN	- Process menu of schedules	134
OPER	- Process CHANGE OPER request	110
REPORT	- Process PRINT REPORT request	104

TABLE 4

MODULES AND THEIR FUNCTIONS (continued)

TABLE 5

MMI DATA BASE

Variable	Function	Values	Set By
DBRDY%	Real-time data base ready flag	0=real time data base not initialized 1=real time data base valid	INITDB, UPDATE
GDTYPE%	Graphics display type indicator	-1=nothing displayed in graphic display area of screen 0=menu displayed 1=process diagram displayed	INITDB, HELP, TOUCHIN, DMENU, CONFIRM, CANCEL, XDD
DIAMEN%	Diagram or menu number currently displayed Menu: 1=diagrams 2=set point limits 3=reports 4=schedules Diagram: 1-NDES% (1 to NDES%) (i=building or DE)	$\geq$ 1 (see GDTYPE%)	DIAGRAM, SETPT, REPORT, SCHED
XTOUCH%	X-coordinate of touch panel input	0-511	TOUCHIN
YTOUCH%	Y-coordinate of touch panel input	0-511	TOUCHIN
NHELPBF%	Number of buffer files in HELP sequence	currently 11	INITDB
STONE%	Flag to determine whether or not to sound the alarm tone	0=do not sound tone 1=sound tone	INITDB, XSE, XSD, XSPL, XAM, ANNUNC
TPAUSE%	Pause counter for alarm tone silencer (pause for two minutes)	0-24 (each increment represents an estimate of five seconds time)	MMI

TABLE 5

MMI DATA BASE (continued)

Variable	Function	Values	Set By
MCOLOR%	Message color flag (for DISMESS) based on message	0=green (cue)	routine which calls DISMESS
	type	1=red (error)	
		-1=white on red blinking (alarm)	
MINDEX%	Record number of message in master file to be displayed by DISMESS	1-NMESS%	Routine which calls DISMESS
NMESS%	Number of messages in DISMESS master file	≥ 0	INITDB
KBTP%	Currently enabled interrupt	0=none	INITDB, HELP, OPER, MLIN
		1=keyboard	MSIN
		2=touch panel	
STYPE%	String type expected for keyboard input	1=single alphabetic (or no) character	any routine which uses KEYIN input
		2=alphabetic string	
		3=numeric string	
NKEYS%	Number of function keys	Currently =10	INITDB
FKLOC% (NKEYS%)	Leftmost (x) coordinate of Ith function key	1-512	INITDB

TABLE 5

MMI DATA BASE (continued)

Variable	Function	Values	Set By
QFCN%	Selected function	0=no key in effect 1=START/ENABLE 2=STOP/DISABLE 3=DISPLAY DIAGRAM 4=SET POINT/LIMITS 5=AUTO 6=PRINT REPORT 7=MODIFY SCHED 8=CHANGE OPER 9=CONFIRM ACTION 10=CANCEL ACTION	INITDB FKEYIN
QSCH%	Selected Schedule	0=None	SCHED
QPT%	Selected point	1-NDES%-schedule 1-NRTPTS%=point	INITDB, GONOGO, SETPT
QMENU%	Selected menu item	≥ 1	INITDB, DIAGRAM, REPORT, MSIN
QAM%	Selected mode	1=auto 0=manual	AUTO
NOPER%	Number of system operators	≥ 1	INITDB
QOPER%	Current operator	1-NOPER%	OPER
OPER$ (NOPER%)	String containing name of Ith operator		INITDB
PASS$ (NOPER%)	String containing password for Ith operator	≥ 1	INITDB
OLEVEL% (NOPER%)	"Security" level of Ith operator	≥ 1	INITDB

TABLE 5

MMI DATA BASE (concluded)

Variable	Function	Values	Set By
NREPORTS%	Number of reports which can be printed	≥ 1	INITDB
LTMP	Temp. storage for lowest non-alarm value		DMENU
HTMP	Temp. storage for highest non-alarm value		DMENU
STMP	Temp. storage for set point		DMENU
ATMP	Temp. storage for alarm point		DMENU
D1%	Argument to DRAWIT indicating type of item	-1=Alarm Indicator 0=Menu Symbol 2=Function key	Any routine which calls DRAWIT
D2%	Argument to DRAWIT indicating which item		Any routine which calls DRAWIT
D3%	Argument to DRAWIT indicating type of change	0=Interior 1=Border 2=Both	Any routine which calls DRAWIT
D4%	Argument to DRAWIT indicating color	1=red 2=green 3=blue 4=white 5=black 6=megenta 7=cyan 8=yellow	Any routine which calls DRAWIT

TABLE 6

REAL-TIME DATA BASE

Variable	Function	Values	Set By
D5%	Argument to DRAWIT indicating rotation	0, 90, 180, or 270 degrees	Any routine which calls DRAWIT
D6%	Argument to DRAWIT containing special infomation	0=none 1=blink 2=stripe	Any routine which calls DRAWIT
SCHTMP(25)	Temporary storage for changing schedule		MENUIN
STOP%	Flag to stop system (shutdown)	1=shutdown 0=do not shutdown	INITDB OPER

TABLE 6

REAL-TIME DATA BASE(continued)

Array	Function	Values	Set By
NDES%	Number of DE's	0-limits of storage	INITDB
NRTPTS%	Number of real-time data points (note: for arrays listed below, I% assumes values of 1 to NRTPTS%)	0-limits of storage	INITDB
ANPT%	Annunciation point (sent to ANNUNC)	1-NRTPTS%	UPDATE
APT%	The current alarm point, as defined by EMCS (never more than one at a time)	0=none 1-NRTPTS%=point	INITDB UPDATE ANNUNC
DPT%	The current disabled point (i.e., operator not yet notified of its status by a message)	0=none 1-NRTPTS%=point	INITDB UPDATE
NEWVAL (NRTPTS%)	Newest value of real-time data point I%	$\geq$ 0.0	UPDATE
DBVAL (NRTPTS%)	Current (previous) value of real-time data point I%	$\geq$ 0.0 (set by INITDB to -9999.0 to signify invalid data)	INITDB UPDATE

TABLE 6

REAL-TIME DATA BASE (continued)

Array	Function	Values	Set By
PAD% (NRTPTS%)	"Analog" vs. "digital" point discriminator, where "analog" refers to a point which can assume a range of values, and "digital" refers to a point which can assume two discrete values representing opposite states	0=analog 1=digital	INITDB
LOWLIM (NRTPTS%)	Lowest non-alarm value for analog point I%	$\geq$ 0.0	INITDB
HILIM (NRTPTS%)	Highest non-alarm value for analog point I%	> LOWLIM (I%)	INITDB
SPT(NRTPTS%)	Set point for analog point I%	$\leq$ HILIM (I%) $\geq$ LOWLIM (I%)	INITDB, XSPL
ALARMVAL (NRTPTS%)	Alarm value for digital point I%		INITDB
ASTAT% (NRTPTS%)	Alarm status flag for real-time data point I%	0=point not in alarm 1=point in alarm	UPDATE
DSTAT% (NRTPTS%)	Disable status flag for real-time data point I% indicates whether point should be processed or ignored	0=point should be processed 1=point is completely disabled; ignore it	INITDB UPDATE
PNAME$ (NRTPTS%)	String array containing name of Ith point for text purposes		INITDB

TABLE 6

REAL-TIME DATA BASE (continued)

Array	Function	Values	Set By
LPOINT% (NRTPTS%)	Number of process diagram on which Ith point is located	≥ 1	INITDB
PTYPE% (NRTPTS%)	Ith point type indicator	0=FID -1=MUX -2=IMUX 1-NRTPTS%=point and type	INITDB
PHIER% (NRTPTS%)	Next higher point to which Ith point is connected in the process hierarchy	0=is highest point N=attached to Nth point	INITDB
PX% (NRTPTS%)	X-coordinate (left) of location of Ith point's symbol on process diagram	1-512	INITDB
PY% (NRTPTS%)	Y-coordinate (lower) of location of Ith point's symbol on process diagram	1-512	INITDB
PROTAT% (NRTPTS%)	Rotation of Ith symbol on process diagram (degrees)	0, 90, 180, or 270	INITDB
TX% (NRTPTS%)	Rightmost touch-sensitive x-coordinate for Ith point	1-512	INITDB
TY% (NRTPTS%)	Uppermost touch-sensitive y-coordinate for Ith point	1-512	INITDB
	NOTE: The points (PX% [I%], PY% [I%, and (TX% [I%], TY% [I%]) define two diagonally opposite corners of the touch-sensitive rectangular zone around point I%.		

TABLE 6

REAL-TIME DATA BASE (concluded)

Array	Function	Values	Set By
AMPT% (NRTPTS%)	Auto/manual flag for Ith point	1=AUTO MODE (normal) 0=MANUAL MODE	INITDB, AUTO
DE% (NDES%)	Auto/manual flag for Ith data environment	1=AUTO MODE (normal) 0=MANUAL MODE	INITDB, AUTO
NDE$(NDES%)	String array containing name of Ith data environment for text purposes		INITDB
S1(25)	Schedule # 1 - Schedule #5		INITDB, XMS

3.2 Module Descriptions

This section includes written descriptions and flowcharts for program modules contained in the MMI software system. The descriptions are arranged in hierarchial order, corresponding to the representation in Figure 12.

NAME: MMI EXEC

PURPOSE:

MMI EXEC serves as the executive program for the system. It is used to supervise and coordinate the receipt and processing of command inputs and real-time data.

OPERATIONAL DESCRIPTION:

MMI EXEC consists of a series of calls to appropriate subroutines which process operator commands and real-time data updates. MMI EXEC also drives the real-time EMCS data update simulator. The simulator emulates operation of an interrupt-driven system.

When MMI EXEC is initiated, it causes the MMI data base to be initialized, runs the HELP display sequence, and gets the simulated initial real-time data base values. Once the system has been initialized, MMI EXEC waits for an interrupt. An interrupt can come from the touch panel, the keyboard, or the EMCS computer simulation. Once an interrupt has been detected, MMI EXEC calls the appropriate subroutine to handle that interrupt. MMI EXEC also performs the "housekeeping chores" of displaying the current time and sounding the alarm tone when appropriate.

CALLED BY:

The program is initiated by the user.

NAME: MMI EXEC (continued)

CALLS:

DISMESS
HELP
INITDB
KEYIN
TOUCHIN
UPDATE

PASSED ARGUMENTS:
(none)

RETURNED ARGUMENTS:
MCOLOR% - message color flag (to DISMESS)
MINDEX% - message index (to DISMESS)

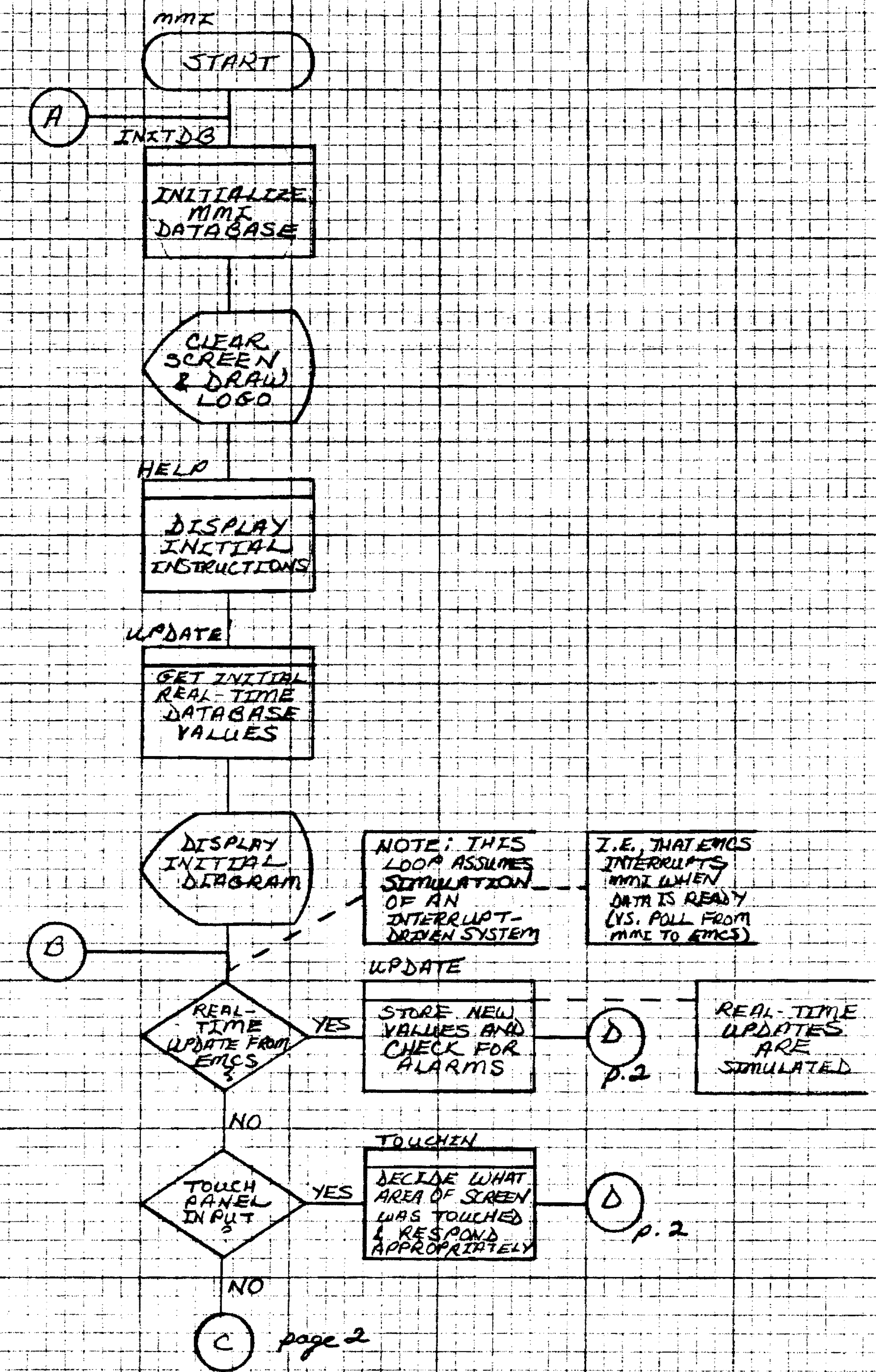
FILE INPUT/OUTPUT:
(none)

HARDWARE INTERACTION:
Touch panel - determine if a "hit" has occurred
Keyboard - determine if keyboard input is attempted

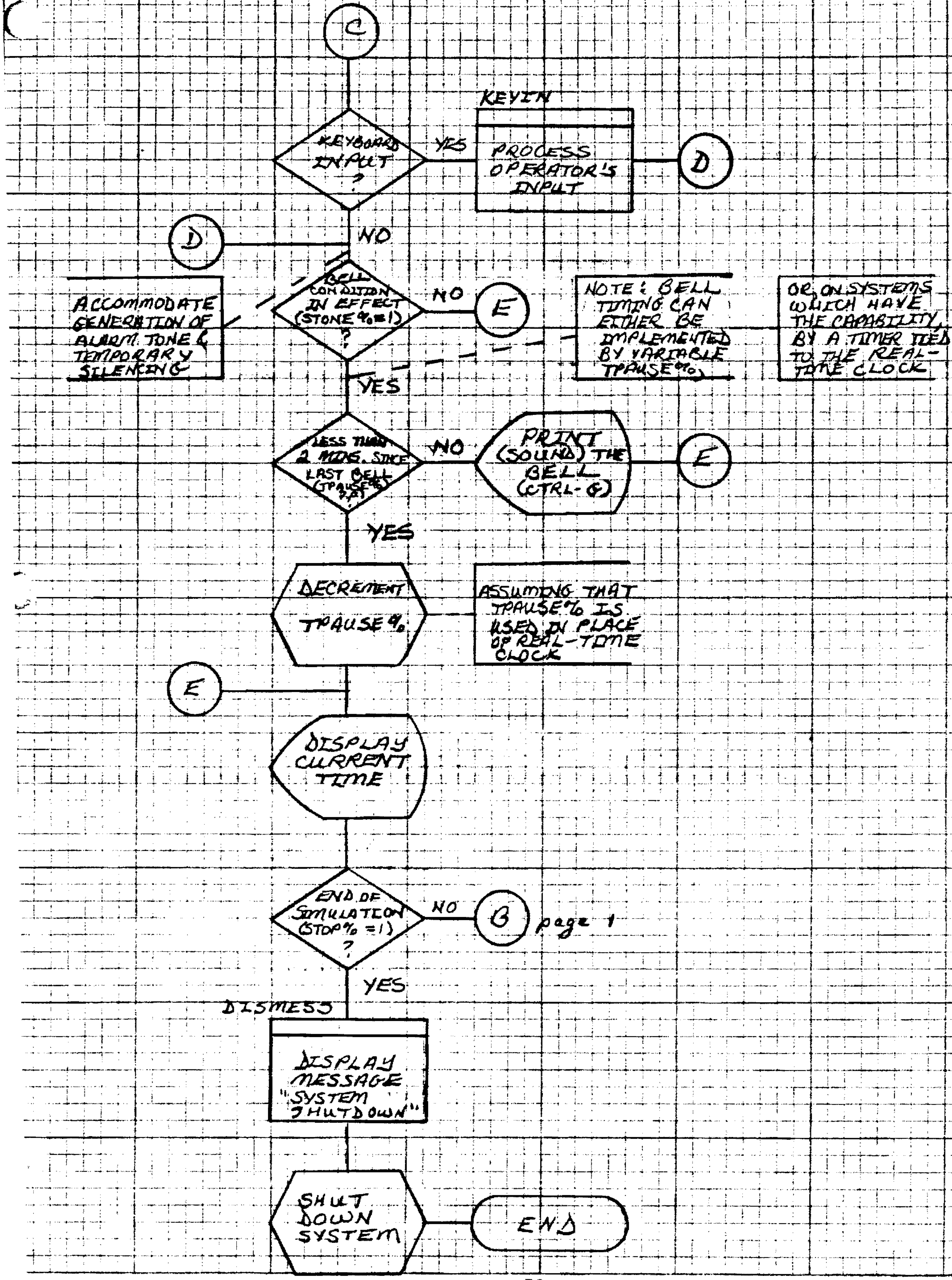
DESIGN NOTES:

The real-time data simulator provides the MMI with real-time data base values in the absence of a link to the EMCS computer. The simulator uses a countdown timer to initiate a random number generator which provides a real-time data "interrupt" (i.e., to signal a change in data). Data values can change randomly, or by simple algorithms to simulate such phenomena as changing temperatures.

MMI: MAN-MACHINE INTERFACE EXECUTIVE



MMI: MAN-MACHINE INTERFACE EXECUTIVE



NAME: INITDB

PURPOSE:

The INITDB subroutine is used to set the initial values of all variables in the MMI data base.

OPERATIONAL DESCRIPTION:

There are two major data bases in the system: the MMI data base contains variables which pertain to software system operation, whereas the real-time data base contains current values which pertain to the (simulated) real-time system. The INITDB subroutine explicitly sets the value of each variable in the man-machine interface data base. INITDB then sets constants (flags, counters, names, etc) for the real-time data base. INITDB also sets the real-time data base ready flag, DBRDY%, to 0 indicate that the real-time data base has not yet been initialized. INITDB then returns to MMI EXEC.

CALLED BY:

MMI EXEC

CALLS:

(none)

PASSED ARGUMENTS:

(none)

NAME: INITDB (continued)

RETURNED ARGUMENTS:

ALARMVAL	KBTP%	PAD%	QPT%
AMPT%	LOWLIM	PASS$	S1
APT%	LPOINT%	PHIER	S2
DPT%	NDE$	PHAME$	S3
DBRDY%	NHELPBF%	PROTAT%	S4
DBVAL	NMESS%	PTYPE%	S5
DE%	NOPER%	PX%	SPT
DSTAT%	NREPORTS%	PY%	STONE%
FKLOC%	NRTPTS%	QAM%	TX%
GDTYPE%	OLEVEL%	QFCN%	TY%
HILIM	OPER$	QMENU%	

(see flowchart for description of initial values)

FILE INPUT/OUTPUT:

(none)

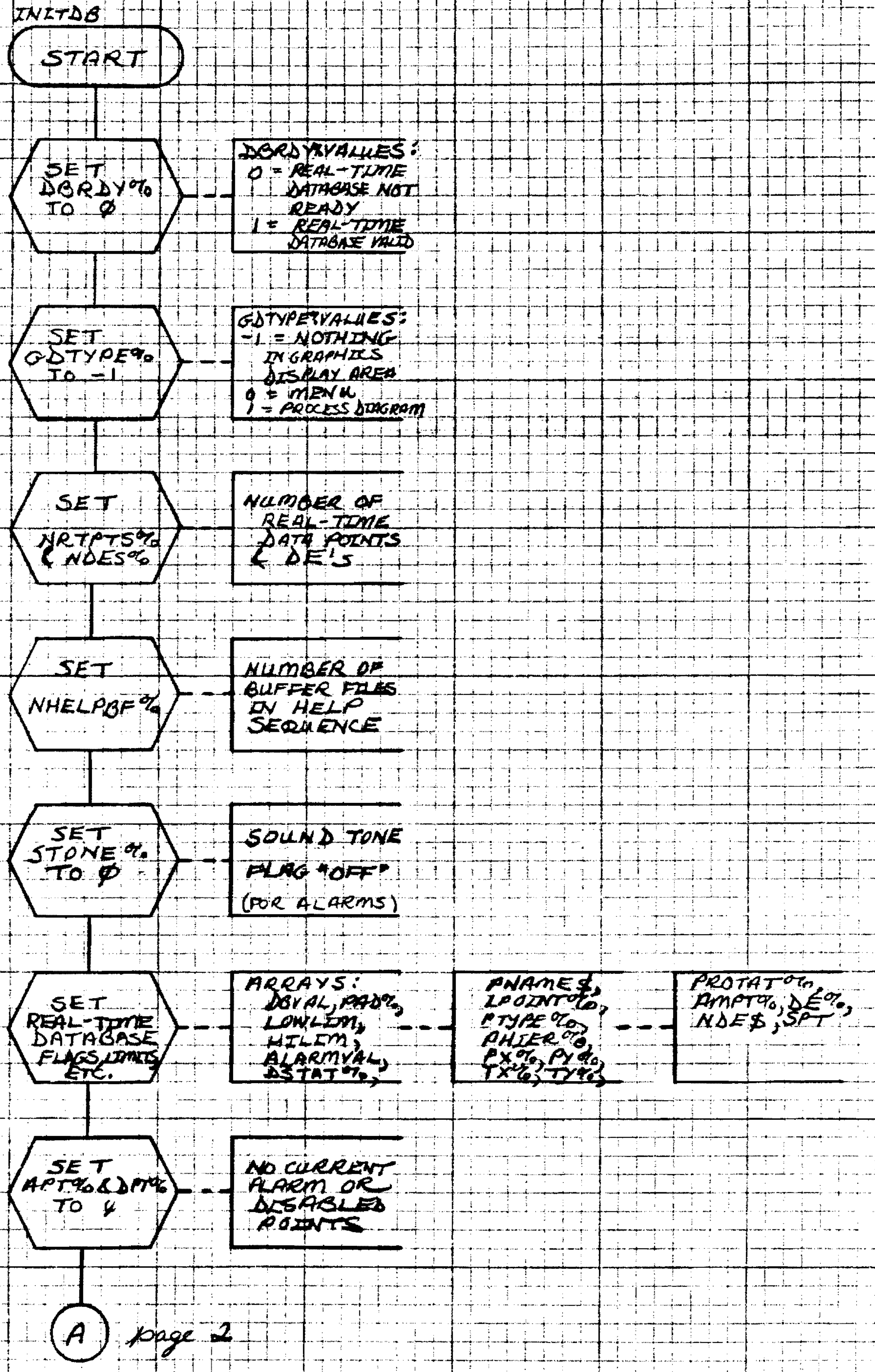
HARDWARE INTERACTION:

(none)

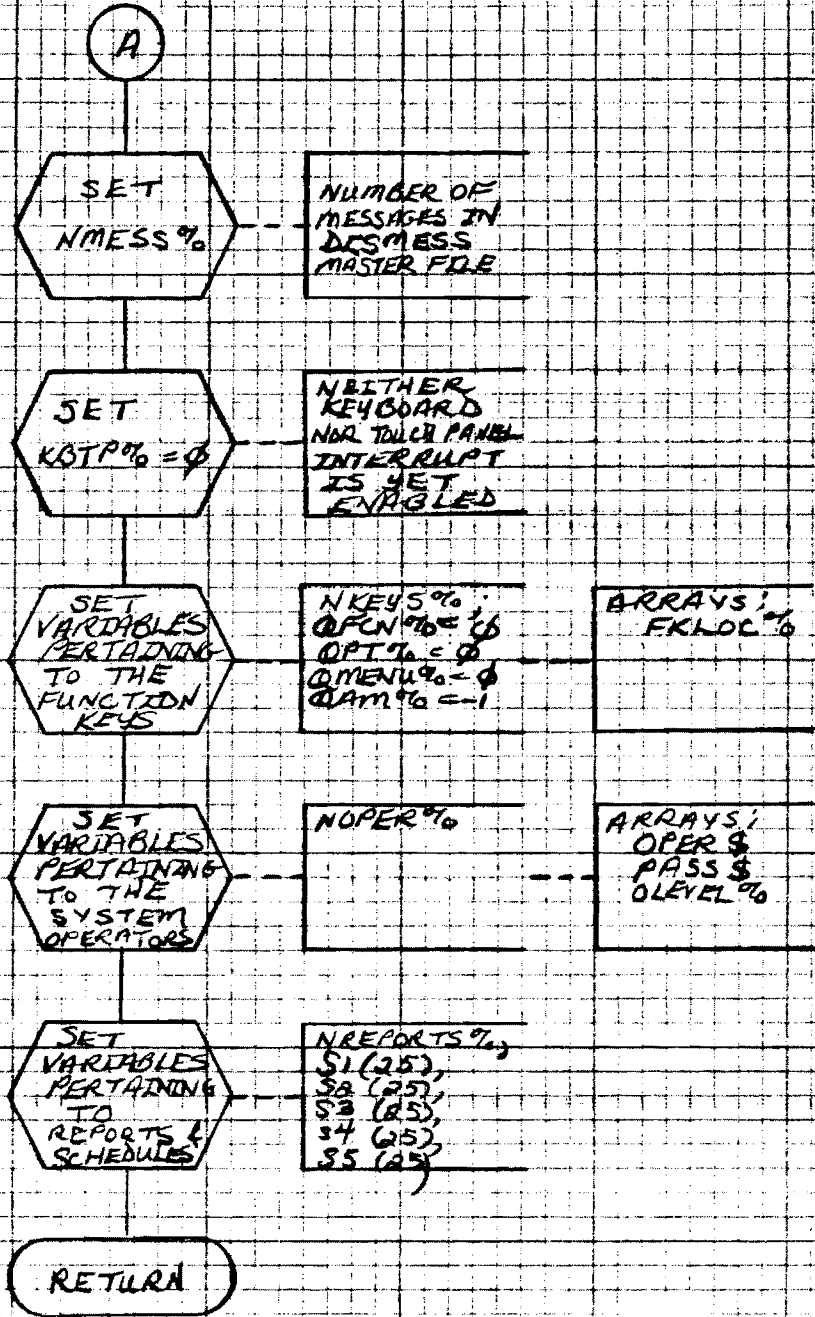
DESIGN NOTES:

(none)

INITDB: INITIALIZE MMI DATABASE



INITDB: INITIALIZE MMI DATABASE



NAME: HELP

PURPOSE:

The HELP subroutine is used to provide a series of screen displays to the operator which contain helpful information describing how to operate the system.

OPERATIONAL DESCRIPTION:

HELP first disables the touch panel and enables the keyboard for input. The variable NHELPBF% contains the number of HELP screen displays available. The subroutine enters a loop which is executed NHELPBF% times. Each time through the loop, a filename is constructed of the form "HLP" + (number of loop), (e.g., HLP1, HLP2, etc.). These filenames correspond to disk files containing the memory image of a HELP display. The file is then read into a memory buffer, from which it is displayed on the screen. The subroutine then waits until the operator presses the <u>Return</u> key before going on to the next display in the sequence. The sequence can be prematurely stopped by typing an <u>S</u> followed by the <u>Return</u> key from the keyboard. When the sequence is complete, the keyboard is disabled and program control is transfered back to the calling routine. If HELP was called by the OPER module, the touch panel is enabled before program control passes back to OPER.

CALLED BY:

MMI Exec at the beginning of system operation.

OPER

CALLS:

KEYIN

NAME: HELP (continued)

PASSED ARGUMENTS:

NHELPBF% - number of HELP messages

RETURNED ARGUMENTS:

KBTP% - currently enabled interrupt

FILE INPUT/OUTPUT:

The subroutine retrieves files of the form HLP**, where ** is an integer in the range 1 through NHELPBF%.

HARDWARE INTERACTION:

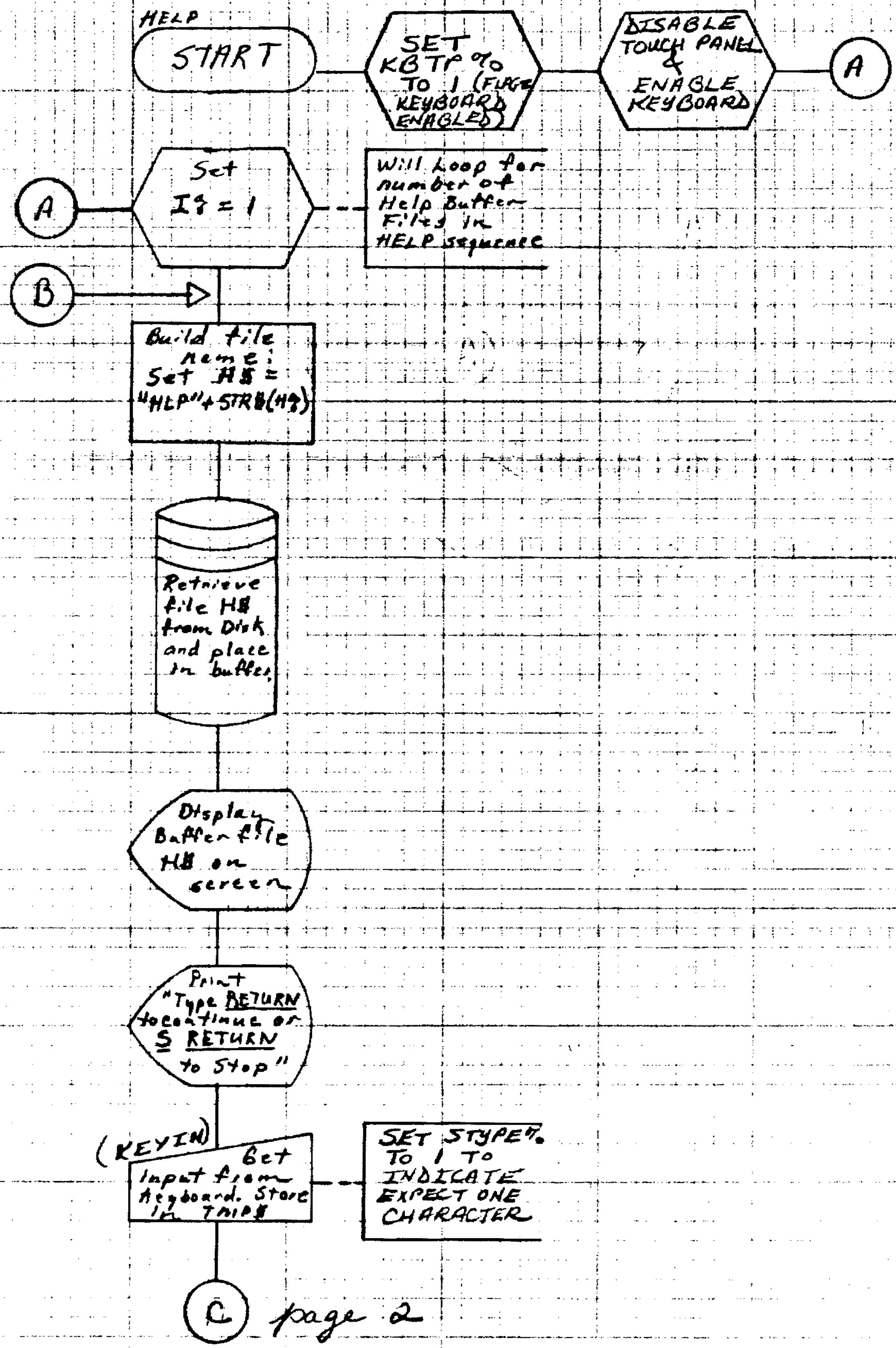
CRT DISPLAY UNIT - used to display information

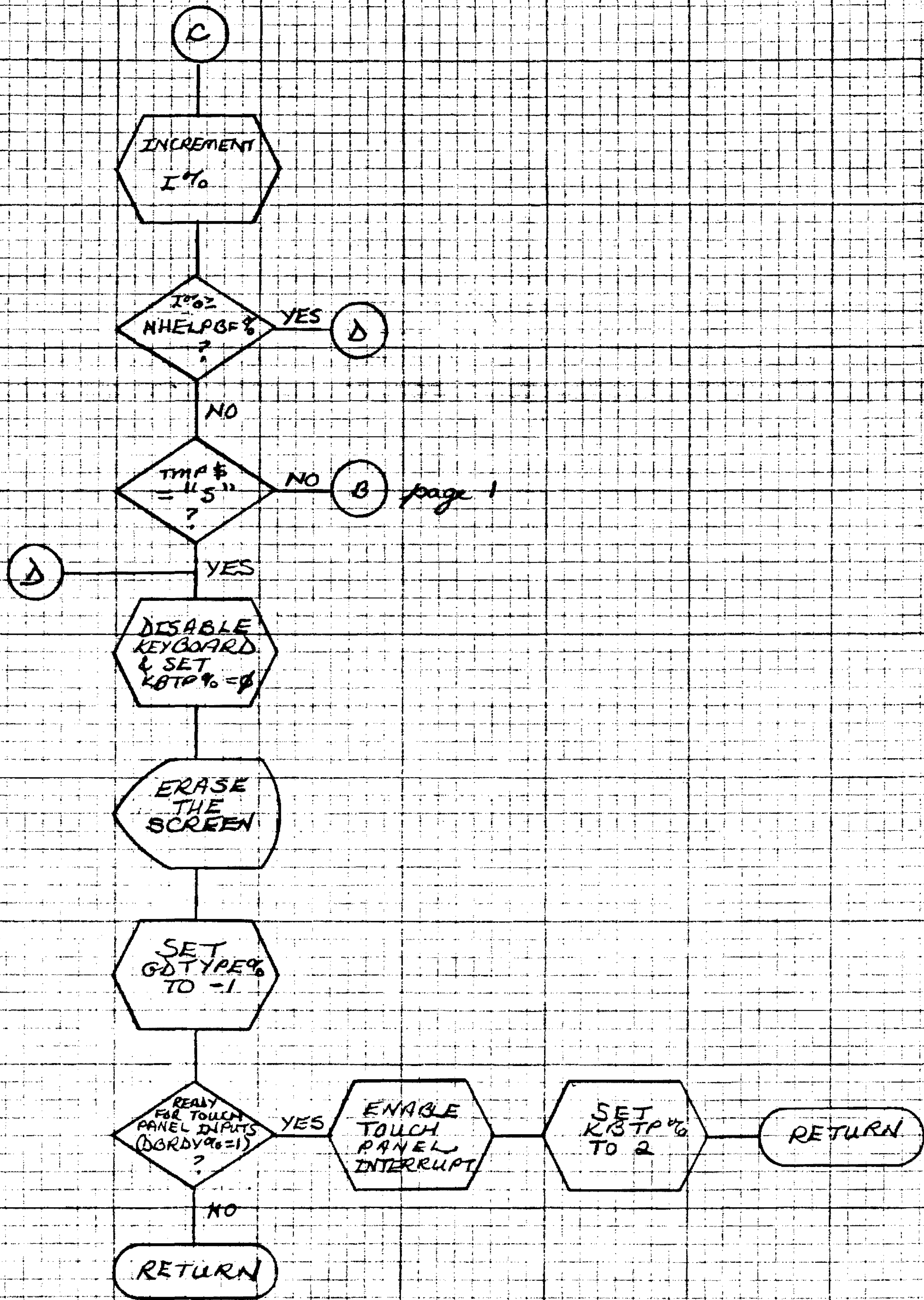
KEYBOARD - used to input operator commands

DESIGN NOTES:

(none)

HELP: PROVIDE OPERATOR INSTRUCTIONS
AND ASSISTANCE




C
INCREMENT I%
I% = NHELPBF% ?
YES
D
NO
TMP$ = "S" ?
NO
B
page 1
D
YES
DISABLE KEYBOARD & SET KBTP% = Ø
ERASE THE SCREEN
SET GDTYPE% TO -1
READY FOR TOUCH PANEL INPUTS (DBRDY% = 1) ?
YES
ENABLE TOUCH PANEL INTERRUPT
SET KBTP% TO 2
RETURN
NO
RETURN

NAME: UPDATE

PURPOSE:

The UPDATE subroutine is used to retrieve new real-time data base values as they are "updated" (i.e., generated by the simulator) and to check them for alarm conditions.

OPERATIONAL DESCRIPTION:

The subroutine gets new values for each point and stores them in an array. UPDATE first checks to see if there is a new point which has been disabled by the automatic control simulator. If so, the point is flagged as disabled and a message to that effect is printed. It then checks each new value to see if it is different from its previous value. If the values are different and the point has not been disabled (in which case it would be ignored), the new value replaces the old value and an alarm check is made. To perform the alarm check, the subroutine first determines whether the point is analog or digital and then compares its value against the alarm value or range specified for that point. If the point is found to be in alarm condition and the alarm has not already been sounded for that point, the ANNUNC subroutine is called to sound the alarm. If the point not is found to be in alarm condition, but was in alarm prior to the receipt of the new value, ANNUNC is called to silence the alarm for that point. If no change in alarm status has occurred, no annunciation is made. When all new values have been read and checked, the real-time data base ready flag (DBRDY%) is set and program control returns to the calling routine.

NAME: UPDATE (continued)

CALLED BY:

MMI EXEC

CALLS:

ANNUNC

PASSED ARGUMENTS:

ALARMVAL - alarm value for digital point
ASTAT% - alarm status flag for real-time data point
DBVAL% - current (previous) value of real-time data point
DPT% - current disabled point
DSTAT% - disable status flag for real-time data point
HILIM - highest non-alarm value for analog point
LOWLIM - lowest non-alarm value for analog point
NRTPTS% - number of real-time data points
PAD% - analog versus digital point discriminator
PNAME$ - textal name of point

RETURNED ARGUMENTS:

APT% - current alarm point
ASTAT% - alarm status flag for real-time point
DBRDY% - real-time data base ready flag
DBVAL - current value of real-time data point

FILE INPUT/OUTPUT:

(none)

NAME: UPDATE (concluded)

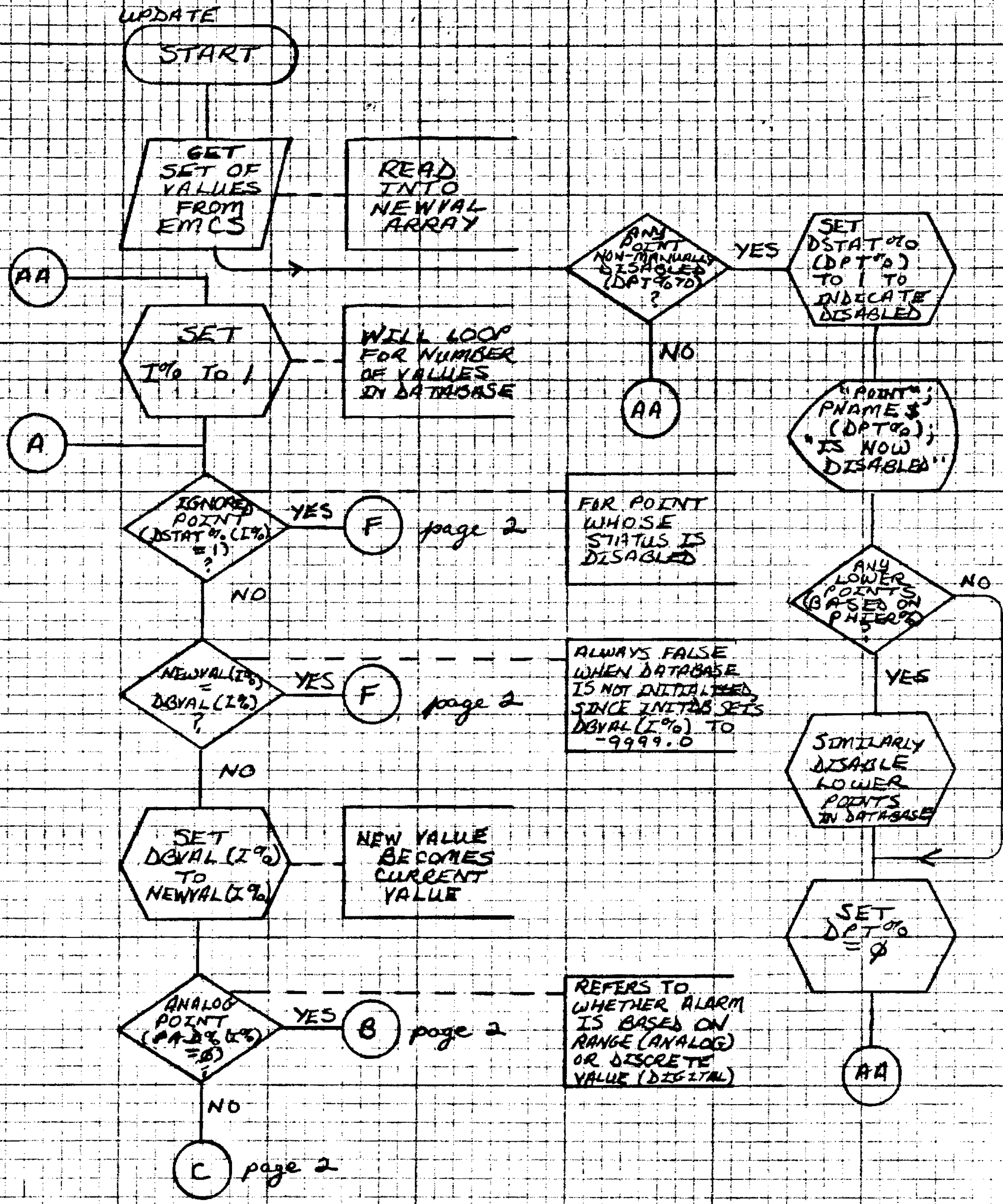
HARDWARE INTERACTION:

(none)

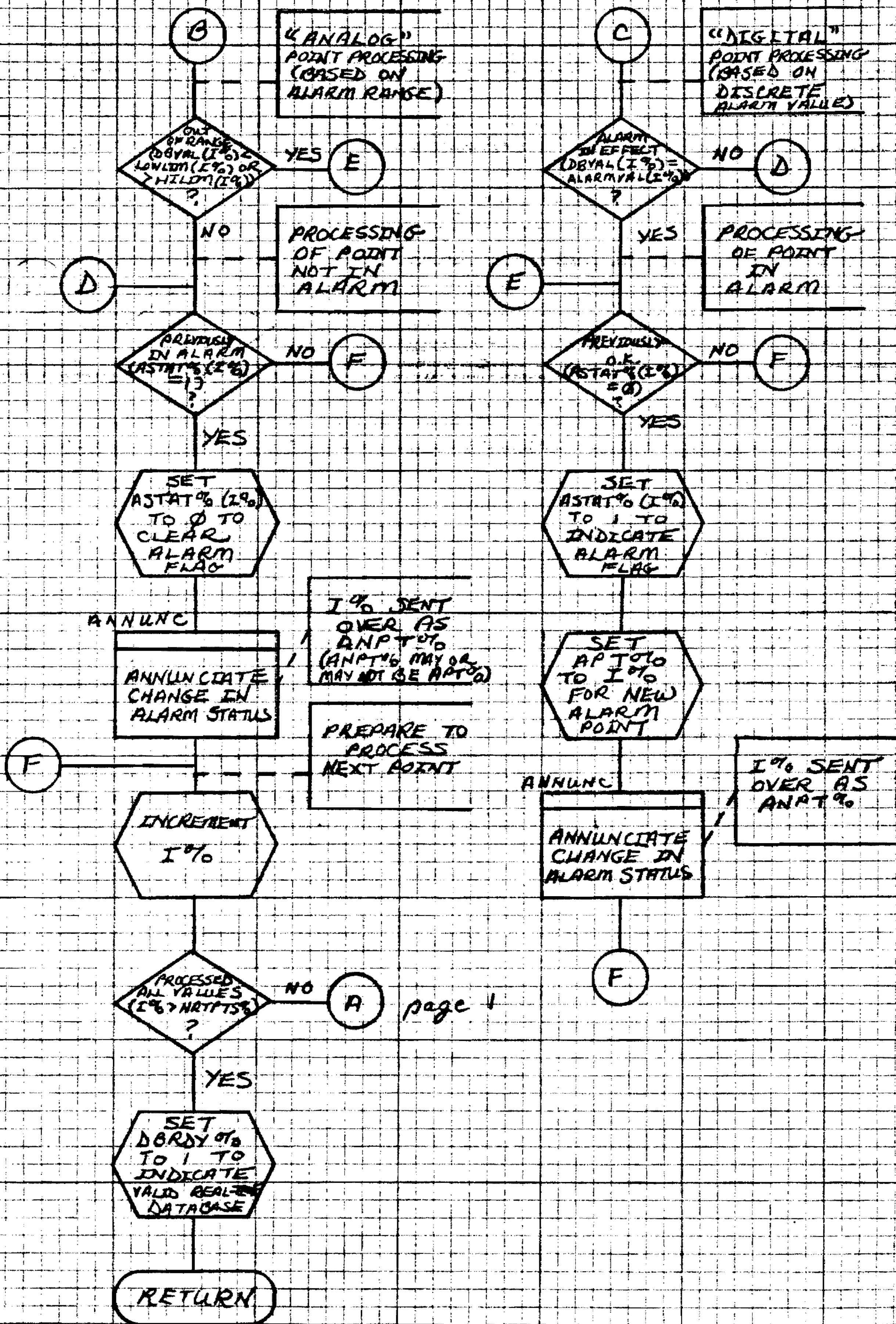
DESIGN NOTES:

(none)

UPDATE: GET REAL-TIME DATABASE VALUES AND CHECK FOR ALARMS



UPDATE: GET REAL-TIME DATABASE VALUES AND CHECK FOR ALARMS



NAME: TOUCHIN

PURPOSE:

The TOUCHIN subroutine is used to get the X & Y coordinates of the point touched on the screen and to call the appropriate subroutine to process the touch request.

OPERATIONAL DESCRIPTION:

The subroutine first obtains the X & Y coordinates of the touch point. It then uses the Y-coordinate to determine which area of the screen was touched, (i.e., alarm indicator, graphics, or function keys). Based on this value and the value of GDTYPE% (which defines whether a menu or process diagram is currently being displayed), the appropriate subroutines are called to process the touch request.

CALLED BY:

MMI EXEC

CALLS:

BUZOFF
FKEYIN
MENUIN
SYMBIN

PASSED ARGUMENTS:

GDTYPE% - graphics display type indicator

RETURNED ARGUMENTS:

(none)

NAME: TOUCHIN (continued)

FILE INPUT/OUTPUT:

(none)

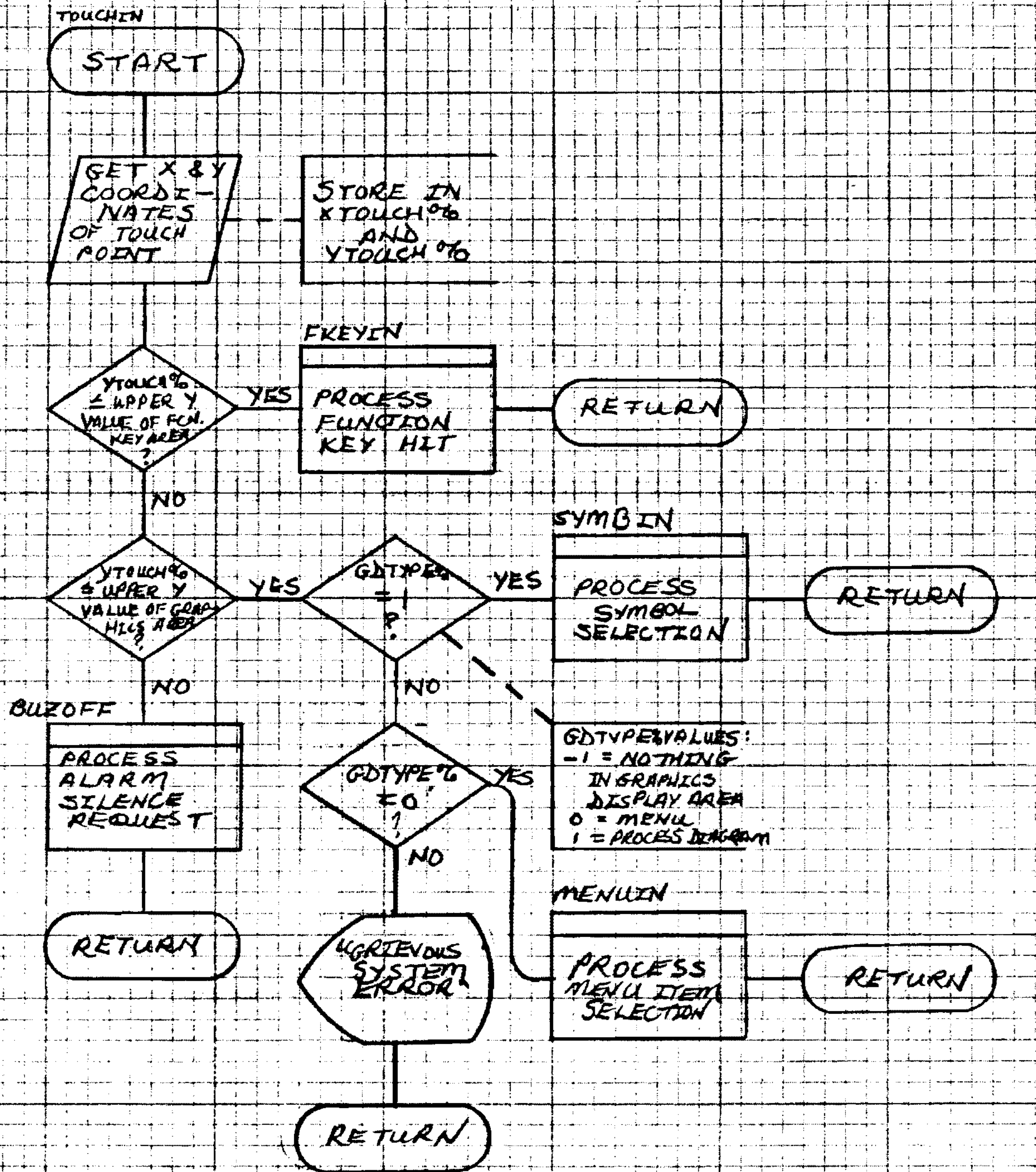
HARDWARE INTERACTION:

Touch Panel - obtain X & Y coordinates of "hit"

DESIGN NOTES:

(none)

TOUCHIN: GET TOUCH PANEL INPUT



NAME: KEYIN

PURPOSE:

The KEYIN subroutine is used to retrieve operator input from the keyboard and to make certain that the input is of the proper type.

OPERATIONAL DESCRIPTION:

The variable TMP$ is used to store a string of input characters. It is first set to the null string. As each character is typed in, it is appended to TMP$. When the Return key is detected, the string is checked for validity based on the value of STYPE%. If the string is invalid, an error message is printed and input is again requested. Otherwise program control returns to the calling routine.

CALLED BY:

MMI EXEC

CALLS:

DISMESS

PASSED ARGUMENTS:

STYPE% - indicates expected string type for input

RETURNED ARGUMENTS:

TMP$ - verified string of input characters.

NAME: KEYIN (continued)

FILE INPUT/OUTPUT:

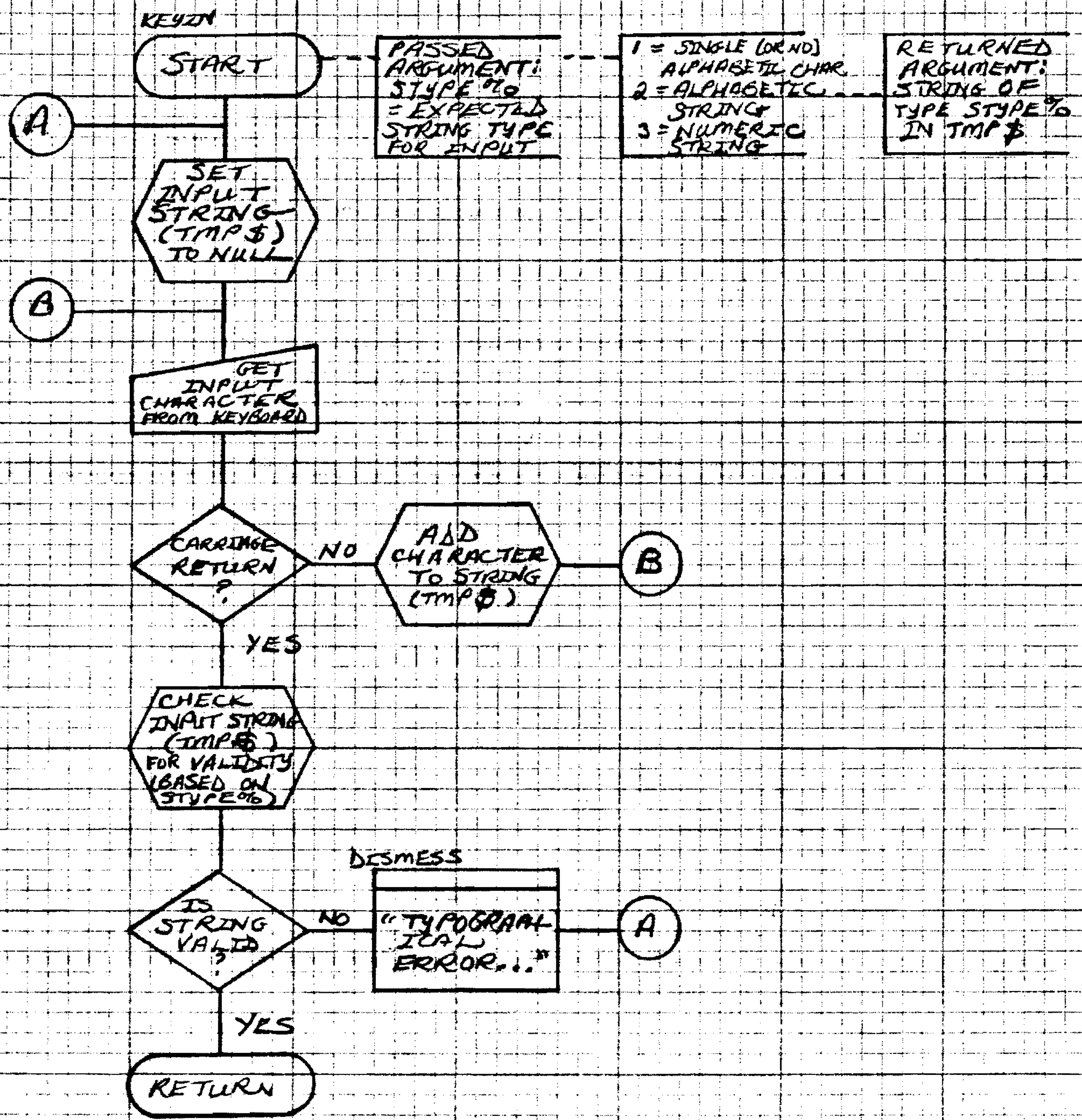
(none)

HARDWARE INTERACTION:

Keyboard - reads input from keyboard.

DESIGN NOTES:

(none)


KEYIN
START
PASSED ARGUMENT: STYPE% = EXPECTED STRING TYPE FOR INPUT
1 = SINGLE (OR NO) ALPHABETIC CHAR
2 = ALPHABETIC STRING
3 = NUMERIC STRING
RETURNED ARGUMENT: STRING OF TYPE STYPE% IN TMP$
A
SET INPUT STRING (TMP$) TO NULL
B
GET INPUT CHARACTER FROM KEYBOARD
CARRIAGE RETURN ?
NO
ADD CHARACTER TO STRING (TMP$)
B
YES
CHECK INPUT STRING (TMP$) FOR VALIDITY (BASED ON STYPE%)
DISMESS
IS STRING VALID ?
NO
"TYPOGRAPHICAL ERROR..."
A
YES
RETURN

NAME: ANNUNC

PURPOSE:

The ANNUNC subroutine is used to make the screen display reflect any change in alarm status, whether it be to signal a new alarm or to turn the alarm off for a device.

OPERATIONAL DESCRIPTION:

The ANNUNC subroutine first decides whether the point is returning from or entering an alarm condition. If the point is returning from alarm, it sets the appropriate flags to turn off the flashing alarm for the device and also turns off the alarm indicator and buzzer. If the point is entering the alarm state, it starts the buzzer and sets the appropriate flags to cause the alarming device's symbol to flash. It also turns on the alarm indicator.

CALLED BY:

UPDATE

CALLS:

DISMESS

DRAWIT

PASSED ARGUMENTS:

ANPT% - annunciation point

APT% - current alarm point

ASTAT% - alarm status flag

DIAMEN% - index number of currently displayed diagram or menu

GDTYPE% - graphics display type indicator

LPOINT% - index number of process diagram on which Ith point is located.

PNAME$ -string array containing textual name of Ith point

NAME: ANNUNC (continued)

RETURNED ARGUMENTS:

APT% - current alarm point

STONE% - flag to determine whether or not to sound the alarm tone

TPAUSE% - pause counter for alarm tone silencer

FILE INPUT/OUTPUT:

(none)

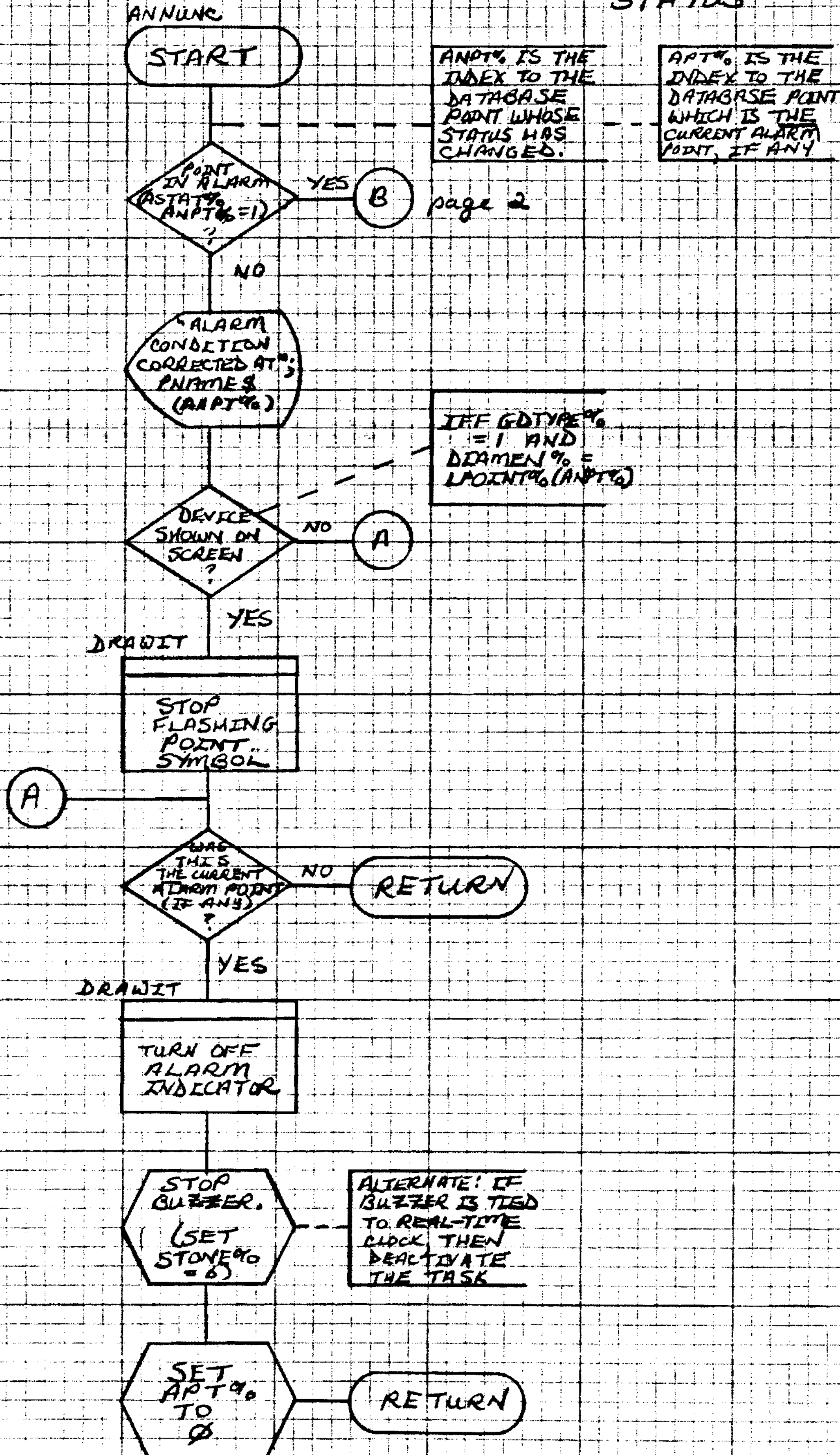
HARDWARE INTERACTION:

CRT DISPLAY - prints messages

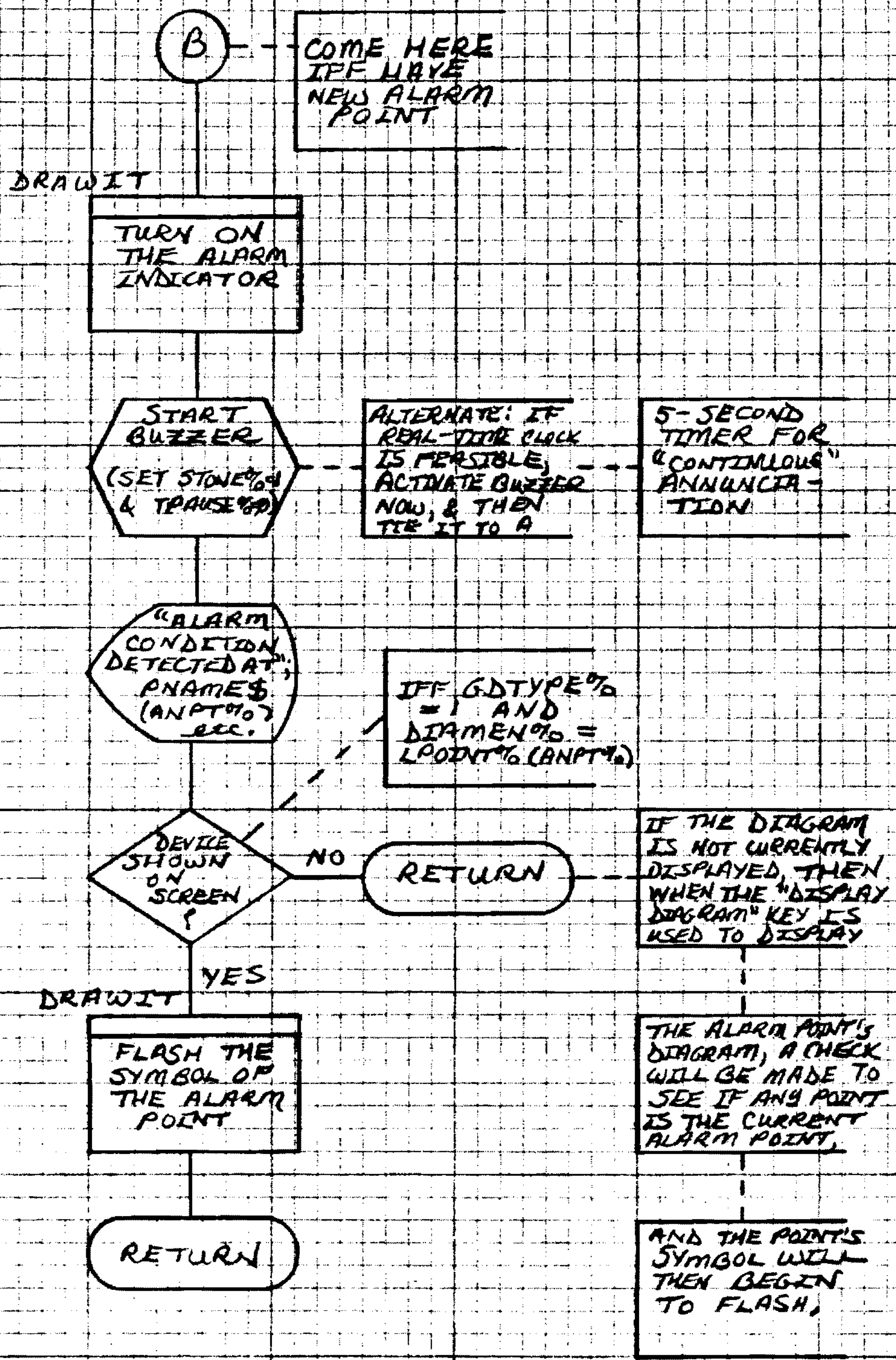
DESIGN NOTES:

When a change in alarm status occurs, it is necessary to cause the appropriate device symbol to flash or stop flashing. The DISPLAY DIAGRAM command software checks alarm status as it draws a diagram, so points in alarm are always represented as flashing symbols on the screen. However, if the diagram containing the alarm point is already displayed, the ANNUNC subroutine must redraw an alarm point symbol so that it either flashes or stops flashing. The variables GDTYPE%, LPOINT%, and DIAMEN% are used by ANNUNC to determine if the diagram containing the alarm point is currently displayed. If GDTYPE% has the value 1, a process diagram is being displayed. DIAMEN% will contain the index number of the diagram currently displayed and LPOINT% will contain the number of the diagram on which the alarm point is located. If GDTYPE% =1 and DIMEN% = LPOINT%, then ANNUNC knows that it must call the DRAWIT subroutine to cause the symbol of the alarm point to start or stop flashing.

ANNUNC: ANNUNCIATE CHANGE IN ALARM STATUS



ANNUNC: ANNUNCIATE CHANGE IN ALARM STATUS



NAME: FKEYIN

PURPOSE:

The FKEYIN subroutine is used to determine which function key has been selected and to call the appropriate subroutine to process the selected function.

OPERATIONAL DESCRIPTION:

First, a check is made to determine whether or not a function key has already been selected. If so, the old function is deselected. Otherwise the X-coordinate of the touch key is used to determine which key has been touched and to call the appropriate subroutine to process the selected function.

CALLED BY:

TOUCHIN

CALLS:

AUTO
CANCEL
CONFIRM
DIAGRAM
DISMESS
DRAWIT
GONOGO
OPER
REPORT
SETPT
SCHED

NAME: FKEYIN (continued)

PASSED ARGUMENTS:

FKLOC%(I) - leftmost (X) coordinate of Ith function key

QFCN% - previous function key selected

XTOUCH% - X coordinate of touch panel input

RETURNED ARGUMENTS:

QFCN% - new function key selection

FILE INPUT/OUTPUT:

(none)

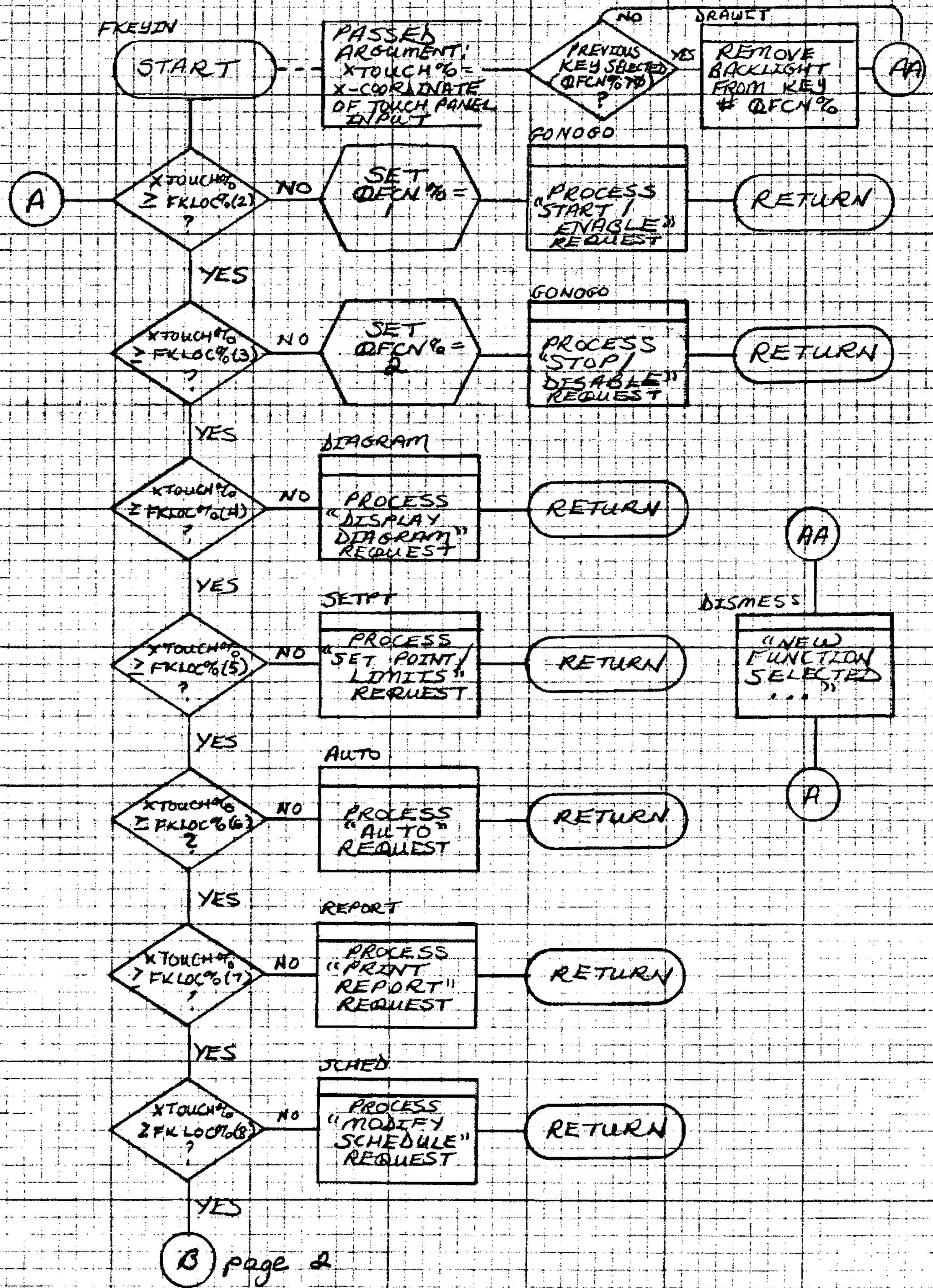
HARDWARE INTERACTION:

(none)

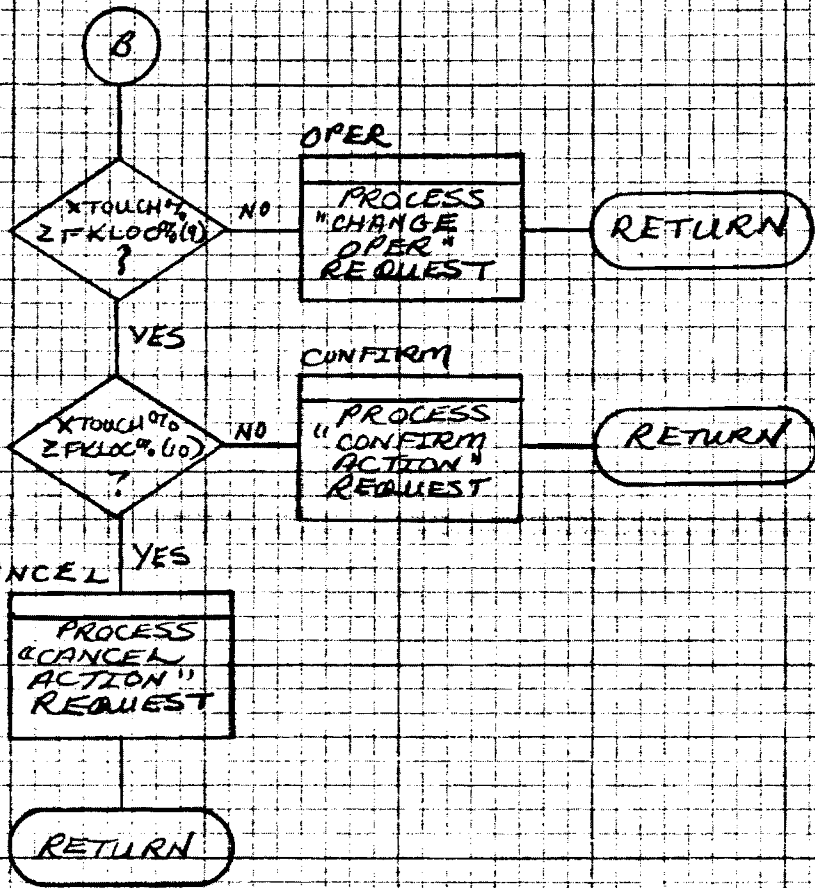
DESIGN NOTES:

(none)

FKEYIN: PROCESS FUNCTION KEY INPUT



FKEYIN: PROCESS FUNCTION KEY INPUT



NAME: SYMBIN

PURPOSE:

The SYMBIN subroutine is used to process device symbol selection requests.

OPERATIONAL DESCRIPTION:

The subroutine first determines which device has been selected and highlights its symbol. If a function has already been selected, checks are made to ensure that conditions are right for execution of the function before control is returned to the calling routine. Appropriate cues and error messages are given.

CALLED BY:

TOUCHIN

CALLS:

DISMESS
DMENU
DRAWIT

PASSED ARGUMENTS:

AMPT% - Auto/manual flag for Ith point
DE% - Data Environment auto/manual mode indicator
DIAMEN%- Number of the diagram or nemu currently displayed
DSTAT% - Disable status flag
NDE$ - Textual name of data environment
NRTPTS%- Number of points in real-time data base
PHIER% - Next higher point in DE hierarchy
PNAME$ - Textual name of point
PTYPE% - Point type indicator

NAME: SYMBIN (continued)

PASSED ARGUMENTS: (concluded)

PX% - leftmost touch-sensitive x-coordinate for Ith point

PY% - Lowermost touch-sensitive Y-coordinate for Ith point

QFCN% - Selected function

QPT% - Selected point

TX% - Rightmost touch-sensitive x-coordinate for Ith point

TY% - Uppermost touch-sensitive y-coordinate for Ith point

XTOUCH%- X-coordinate of touch "hit"

YTouch%- Y-coordinate of touch "hit"

RETURNED ARGUMENTS:

QAM% - Selected mode

QPT% - Selected point

FILE INPUT/OUTPUT:

(none)

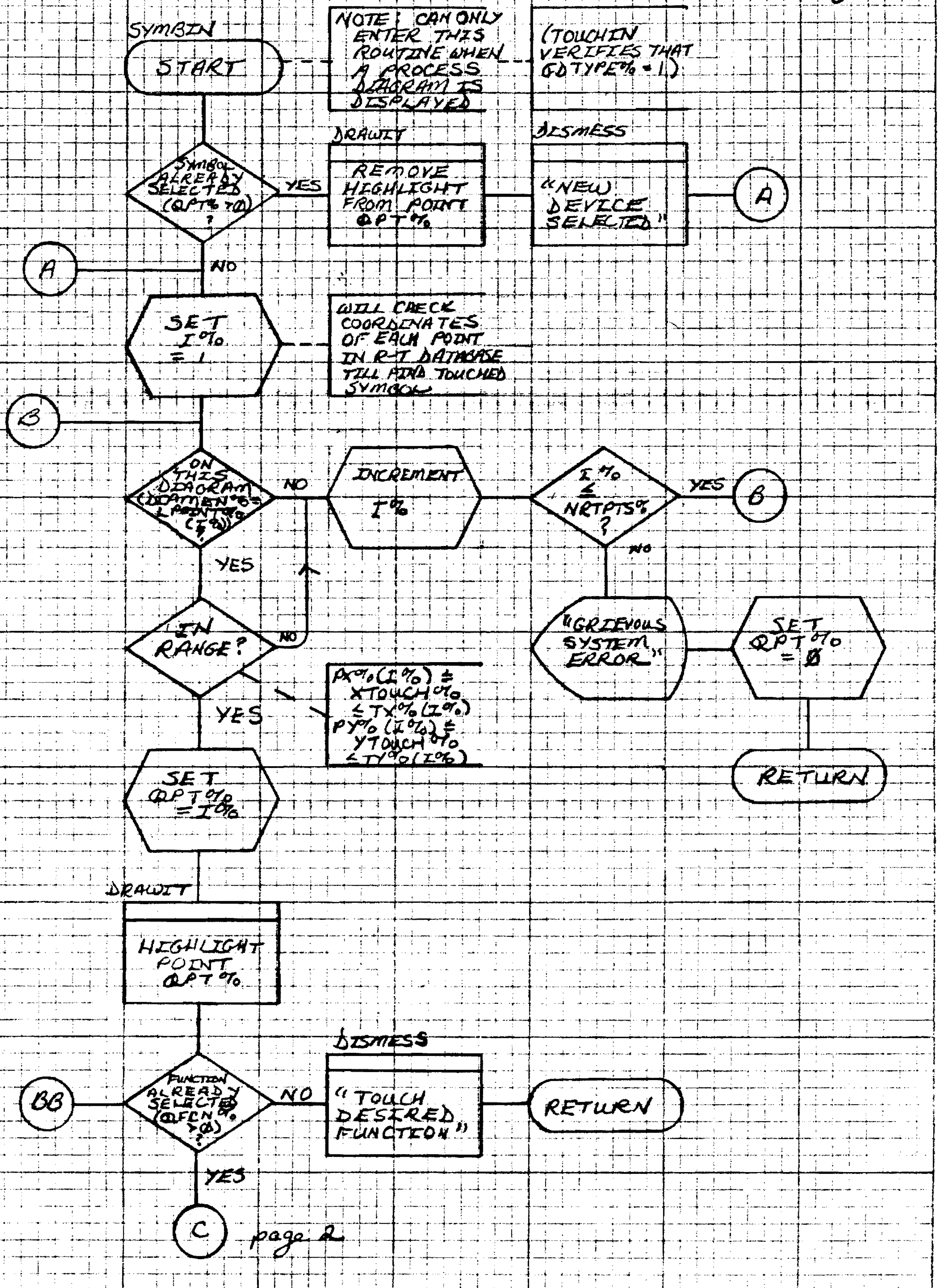
HARDWARE INTERACTION:

CRT DISPLAY - Display messages

DESIGN NOTES:

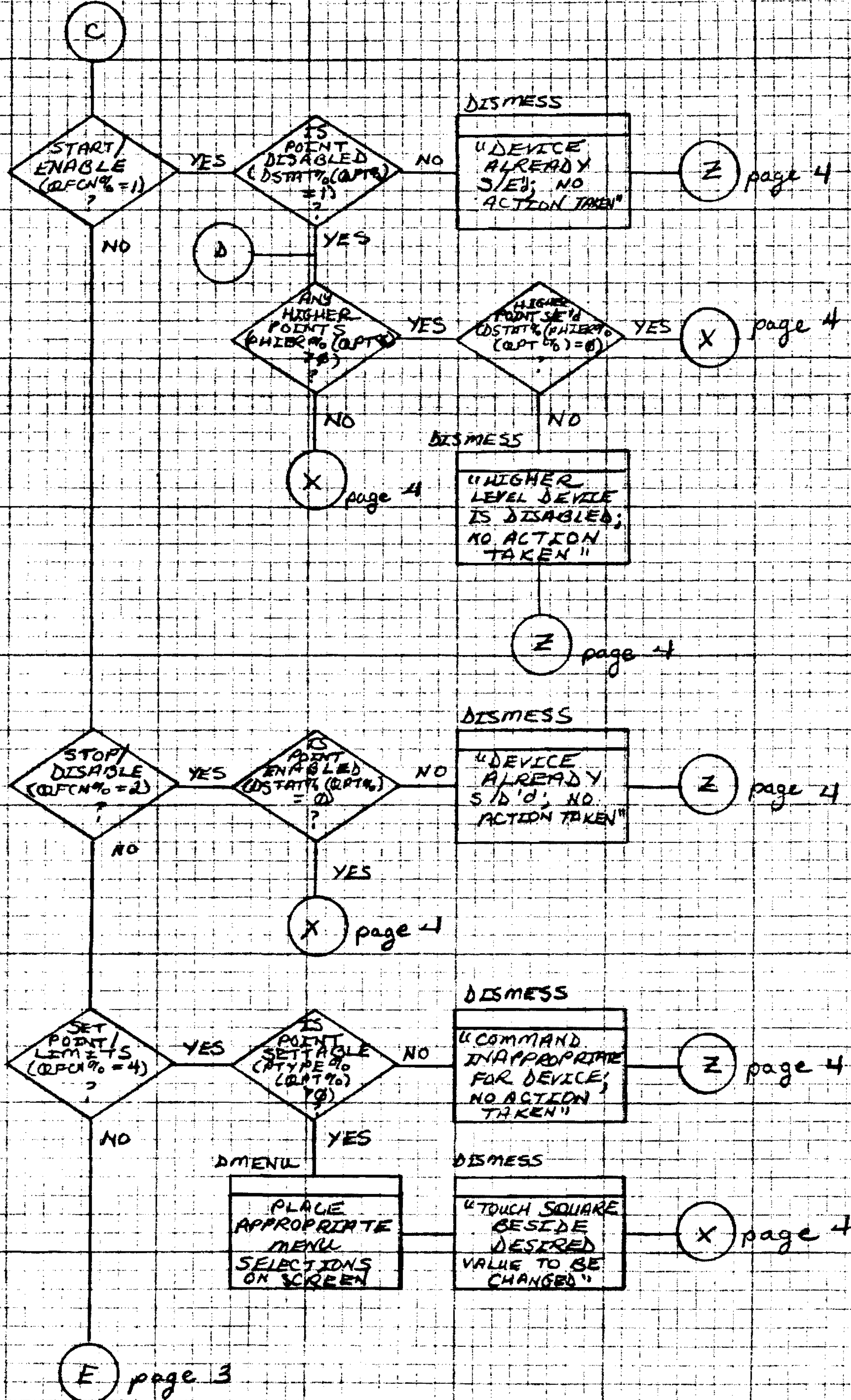
(none)

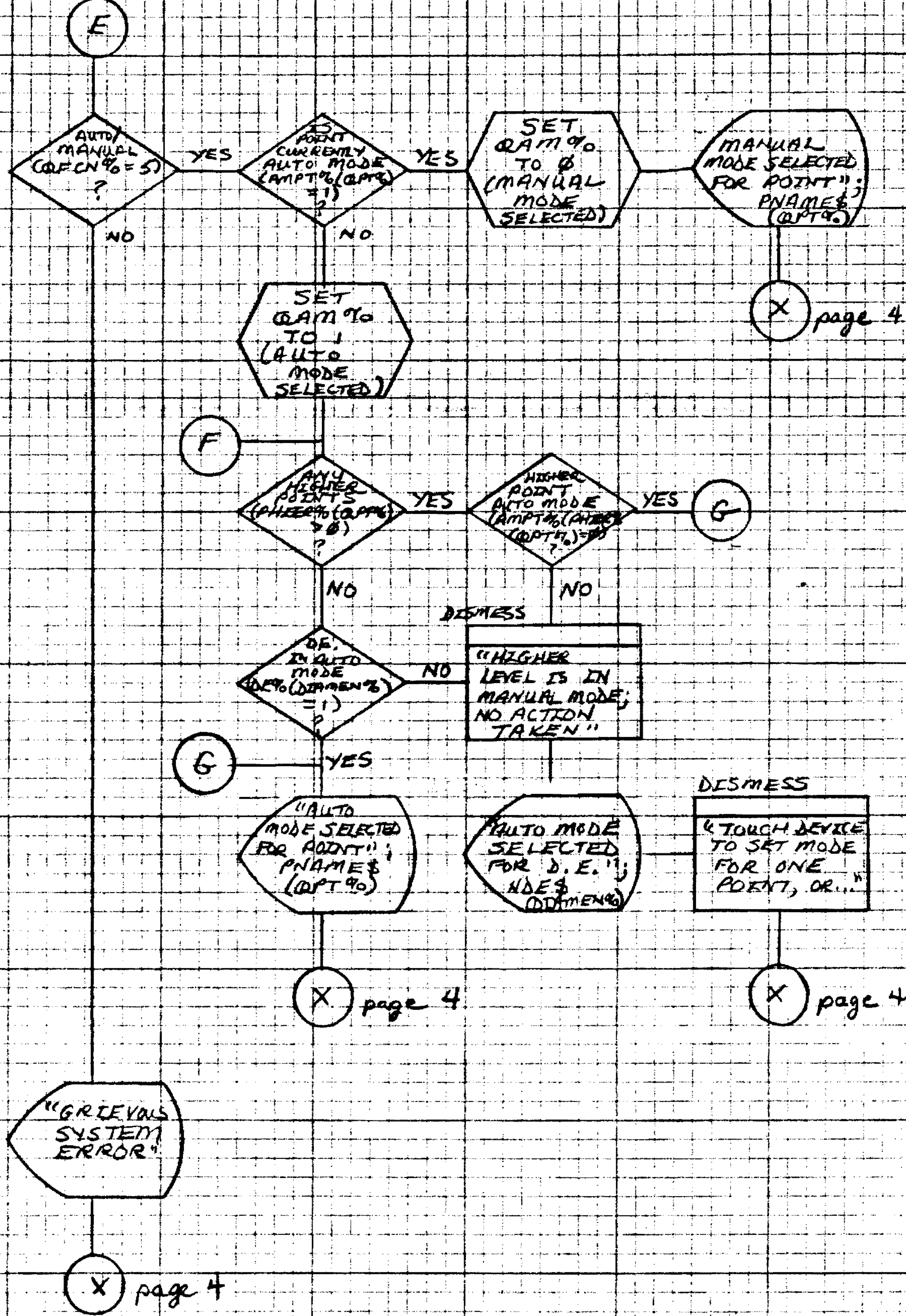
SYMBIN: PROCESS SYMBOL SELECTION



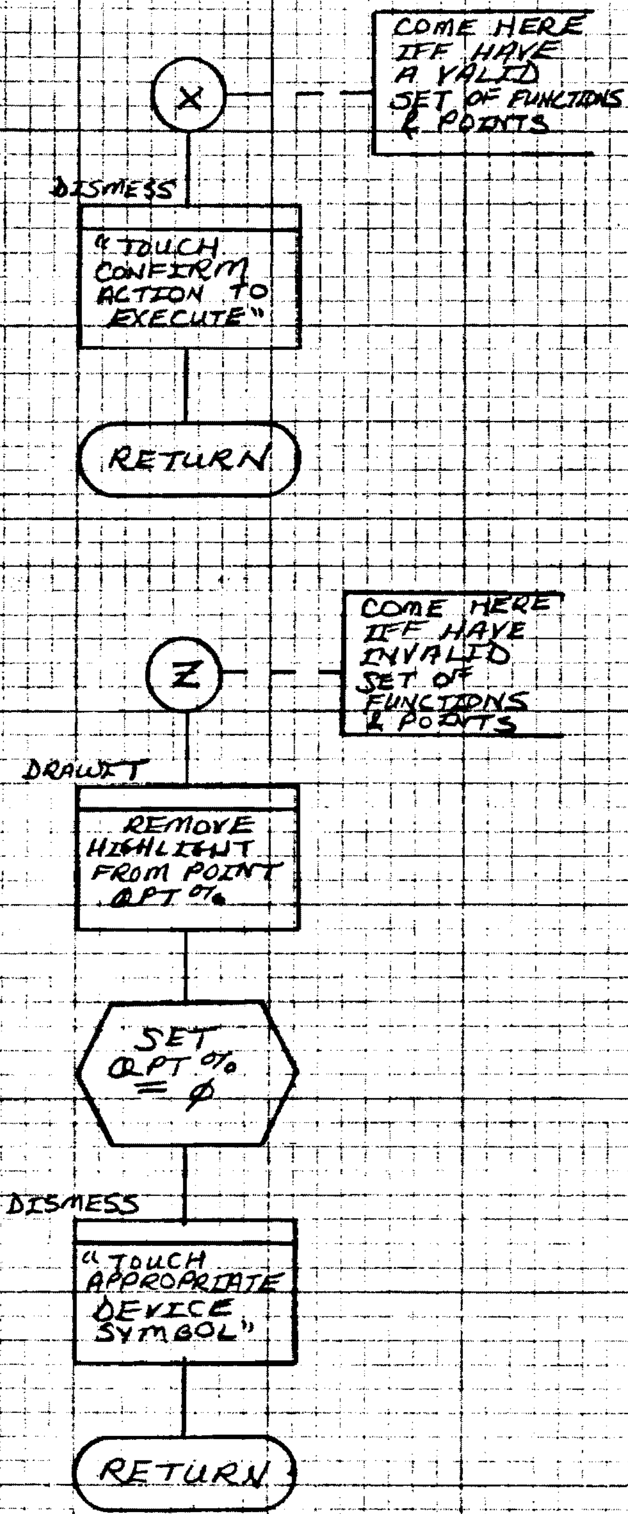
page 2

SYMBIN: PROCESS SYMBOL SELECTION




E
AUTO/MANUAL (@FCN%=5) ?
YES
NO
IS POINT CURRENTLY AUTO MODE (AMPT%(@PT%)=1) ?
YES
NO
SET @AM% TO Ø (MANUAL MODE SELECTED)
"MANUAL MODE SELECTED FOR POINT"; PNAME$(@PT%)
X page 4
SET @AM% TO 1 (AUTO MODE SELECTED)
F
ANY HIGHER POINTS (PHIER%(@PT%)>Ø) ?
YES
NO
HIGHER POINT AUTO MODE (AMPT%(PHIER%(@PT%))=1) ?
YES
G
NO
DISMESS
"HIGHER LEVEL IS IN MANUAL MODE; NO ACTION TAKEN"
D.E. IN AUTO MODE (DE%(DIAMEN%)=1) ?
NO
YES
G
"AUTO MODE SELECTED FOR POINT"; PNAME$(@PT%)
X page 4
"AUTO MODE SELECTED FOR D.E."; NDE$(DIAMEN%)
DISMESS
"TOUCH DEVICE TO SET MODE FOR ONE POINT, OR..."
X page 4
"GRIEVOUS SYSTEM ERROR"
X page 4

SYMBIN: PROCESS SYMBOL SELECTION



NAME: MENUIN

PURPOSE:

The MENUIN subroutine is used to process menu item selection.

OPERATIONAL DESCRIPTION:

The MENUIN subroutine uses the DIAMEN% variable to determine which menu is to be processed. It then calls the appropriate subroutine to process the particular menu.

CALLED BY:

TOUCHIN

CALLS:

MDIN
MLIN
MRIN
MSIN

PASSED ARGUMENTS:

DIAMEN% - number of diagram or menu currently displayed

RETURNED ARGUMENTS:

(none)

FILE INPUT/OUTPUT:

(none)

NAME: MENUIN (continued)

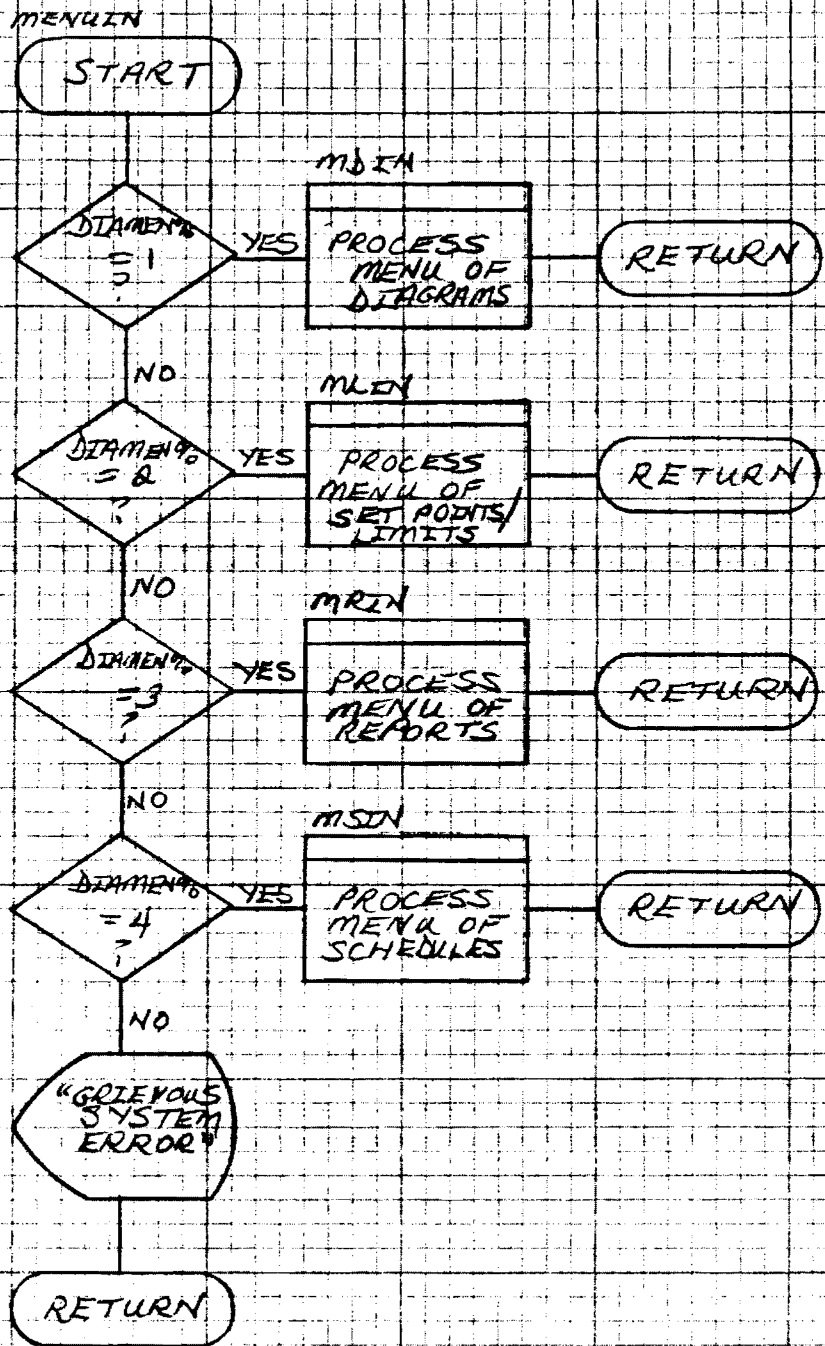
HARDWARE INTERACTION:

(none)

DESIGN NOTES:

(none)

MENUIN: PROCESS MENU ITEM SELECTION



NAME: BUZOFF

PURPOSE:

The purpose of the BUZOFF subroutine is to process the operator alarm silence request.

OPERATIONAL DESCRIPTION:

BUZOFF sets the variable TPAUSE% to the value 24. This value is an estimate of the number of passes through MMI EXEC which can be completed in two minutes. TPAUSE% is decremented each time control passes through MMI EXEC. The audible alarm will not sound until the value of TPAUSE% has reached zero. BUZOFF displays a message to the operator stating that it has acknowledged this request.

CALLED BY:

TOUCHIN

CALLS:

DISMESS

PASSED ARGUMENTS:

(none)

RETURNED ARGUMENTS:

TPAUSE% - pause counter for alarm tone silencer

FILE INPUT/OUTPUT:

(none)

NAME: BUZOFF (continued)

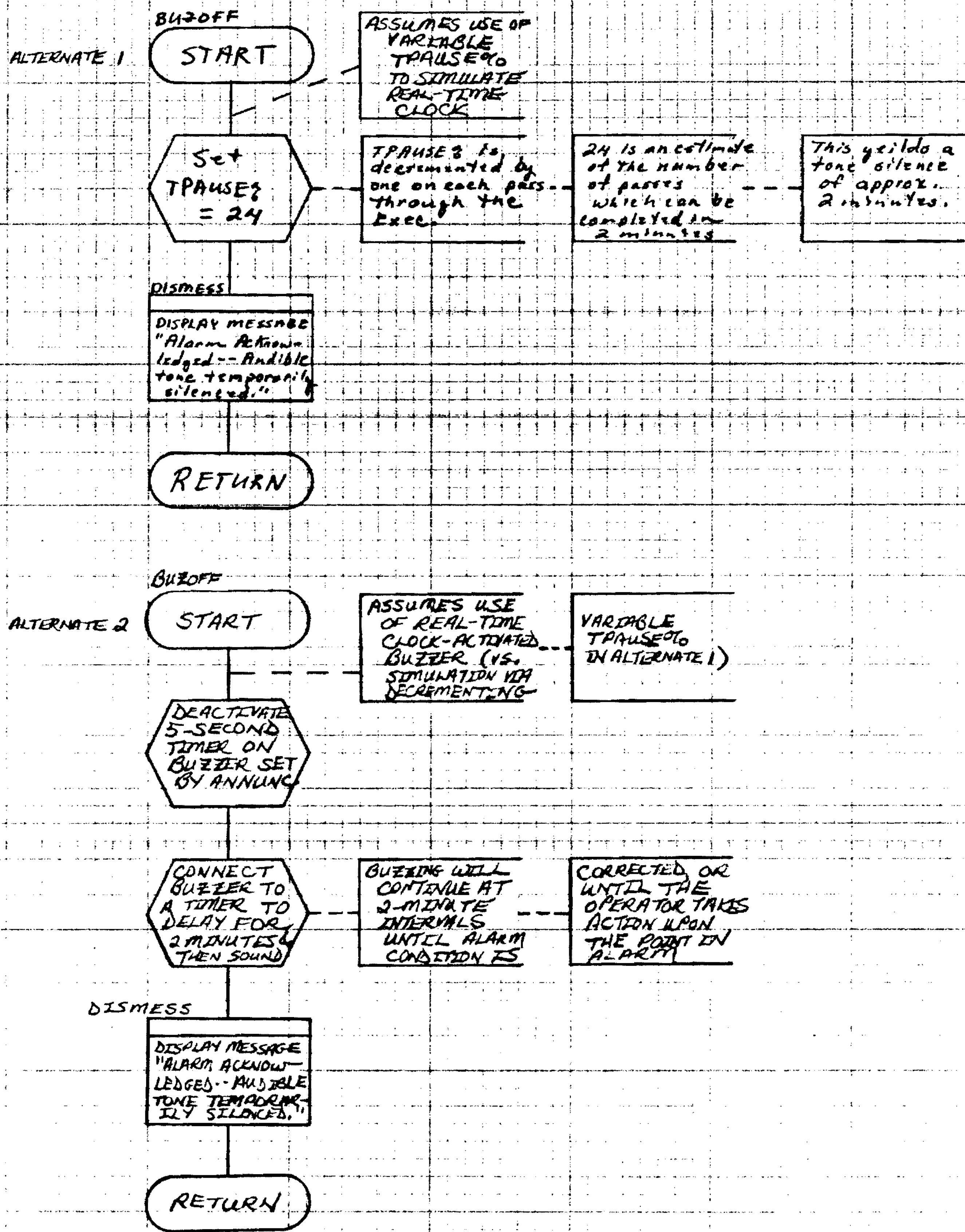
HARDWARE INTERACTION:

(none)

DESIGN NOTES:

An alternate approach would be to tie the alarm pause timer to the real time clock. The design will be modified to incorporate use of the real time clock should this alternative appear feasible.

BUZOFF: PROCESS ALARM SILENCE REQUEST



NAME: GONOGO

PURPOSE:

The GONOGO subroutine is used to process the START/ENABLE and STOP/DISABLE command requests. It does not actually execute these functions, but rather ensures that conditions are right for their execution.

OPERATIONAL DESCRIPTION:

The GONOGO subroutine backlights the appropriate function key and checks to make sure that a process diagram is currently displayed. If not, an error message is given. Otherwise, it checks to see if a device has been selected. If not then program control passes to the calling routine. If both a function and device have been selected, a check is made to insure that the point can be operated upon. If so then appropriate cues are given and program control returns to the calling routine. If the function cannot be executed upon the selected point, the point is deselected and appropriate messages are given.

CALLED BY:

FKEYIN

CALLS:

DISMESS

DRAWIT

NAME: GONOGO (continued)

PASSED ARGUMENTS:

DSTAT% - Disable flag

GDTYPE% - Graphics display type indicator

PHIER% - Next higher point in process diagram hierarchy

QFCN% - Function selected

QPT% - Selected point

RETURNED ARGUMENTS:

QFCN% - Selected function

QPT% - Selected point

FILE INPUT/OUTPUT:

(none)

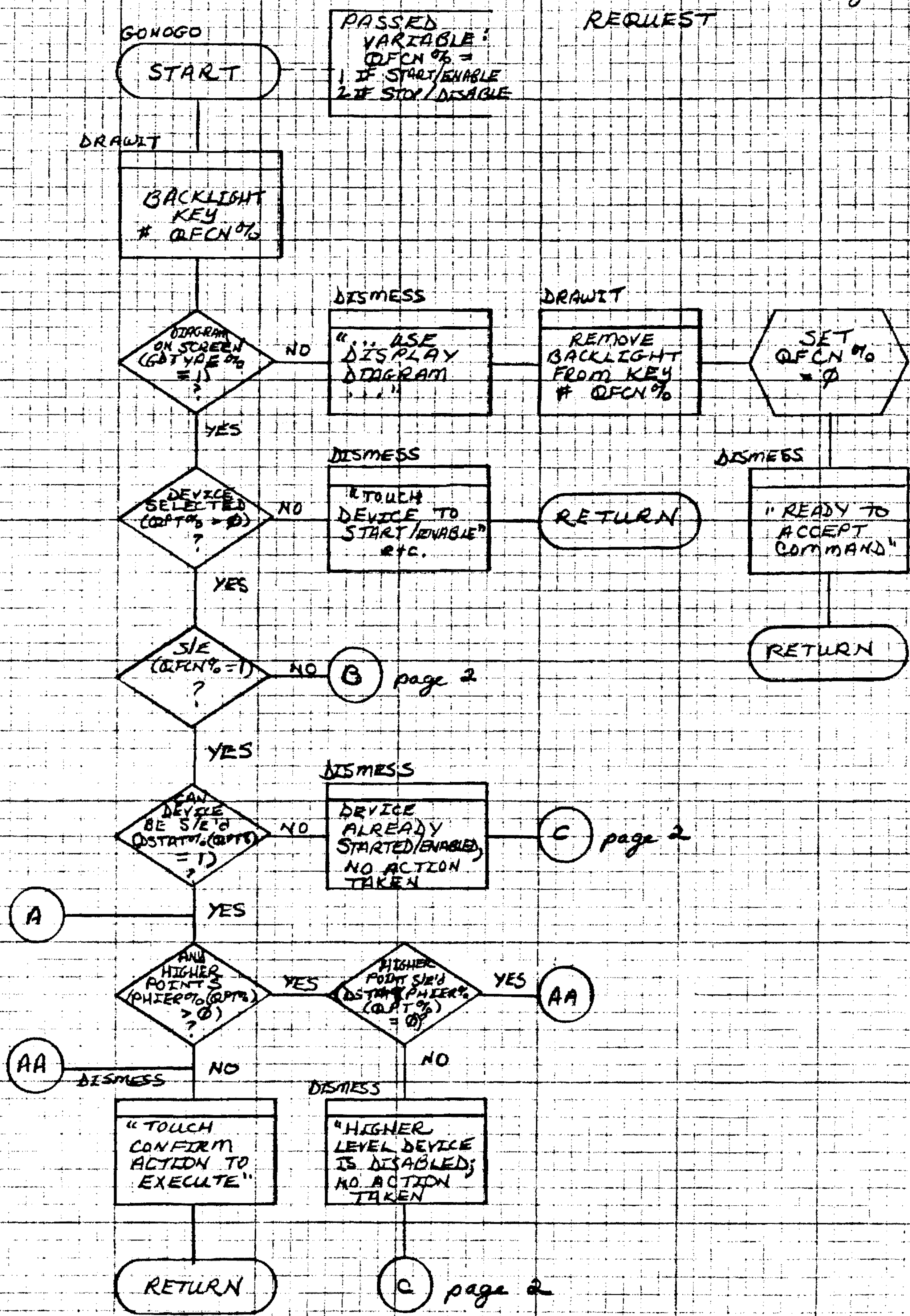
HARDWARE INTERACTION:

(none)

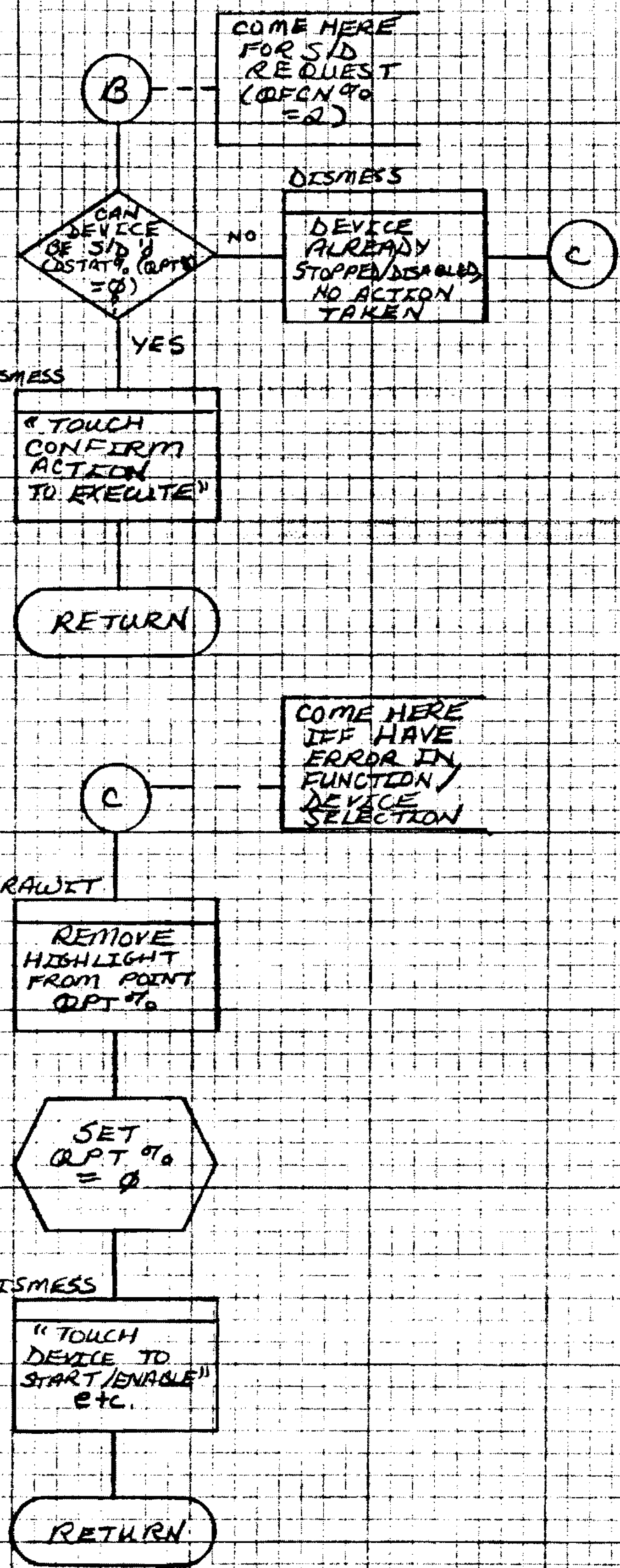
DESIGN NOTES:

(none)

GONOGO: PROCESS "START/ENABLE" OR "STOP/DISABLE" REQUEST



GONOGO: PROCESS "START/ENABLE" OR "STOP/DISABLE" REQUEST



NAME: DIAGRAM

PURPOSE:

The DIAGRAM subroutine is used to process the DISPLAY DIAGRAM command request.

OPERATIONAL DESCRIPTION:

The DISPLAY DIAGRAM function key is backlighted and the appropriate control variables are initialized. The DMENU subroutine is called to display the available DE selections.

CALLED BY:

FKEYIN

CALLS:

DISMESS
DMENU
DRAWIT

PASSED ARGUMENTS:

(none)

RETURNED ARGUMENTS:

DIAMEN% - Number of diagram or menu currently displayed
QFCN% - Selected function
QMENU% - Selected menu item
QPT% - Selected point

NAME: DIAGRAM (continued)

FILE INPUT/OUTPUT:

(none)

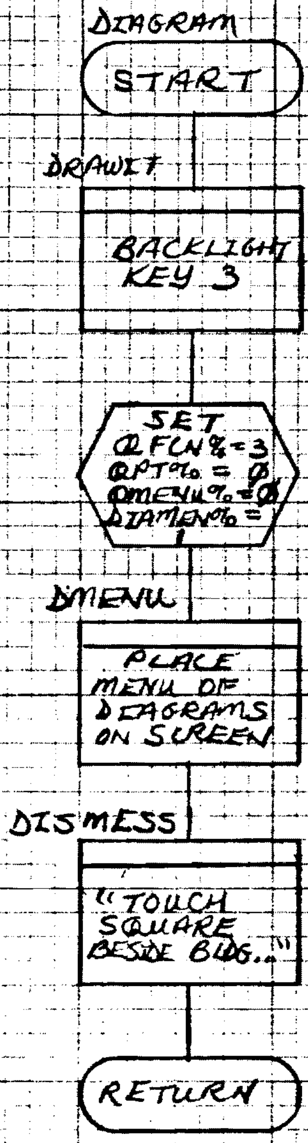
HARDWARE INTERACTITON:

(none)

DESIGN NOTES:

(none)

DIAGRAM: PROCESS "DISPLAY DIAGRAM" REQUEST



NAME: SETPT

PURPOSE:

Process the SET POINT/LIMITS command request. SETPT does not actually execute the command, but rather ensures that conditions are right for execution.

OPERATIONAL DESCRIPTION:

The SET POINT/LIMITS function key is backlighted and checks are made to ensure that a process diagram is on the screen and that, if a point has been selected, it is of the appropriate type. Appropriate cues and error messages are given.

CALLED BY:

FKEYIN

CALLS:

DISMESS
DMENU
DRAWIT

PASSED ARGUMENTS:

GDTYPE% - Graphics display type indicator
PTYPE% - Type of point
QPT% - Selected point

RETURNED ARGUMENTS:

DIAMEN% - Number of diagram or menu currently displayed
QFCN% - Selected function
QMENU% - Selected menu item
QPT% - Selected point

NAME: SETPT (continued)

FILE INPUT/OUTPUT:

(none)

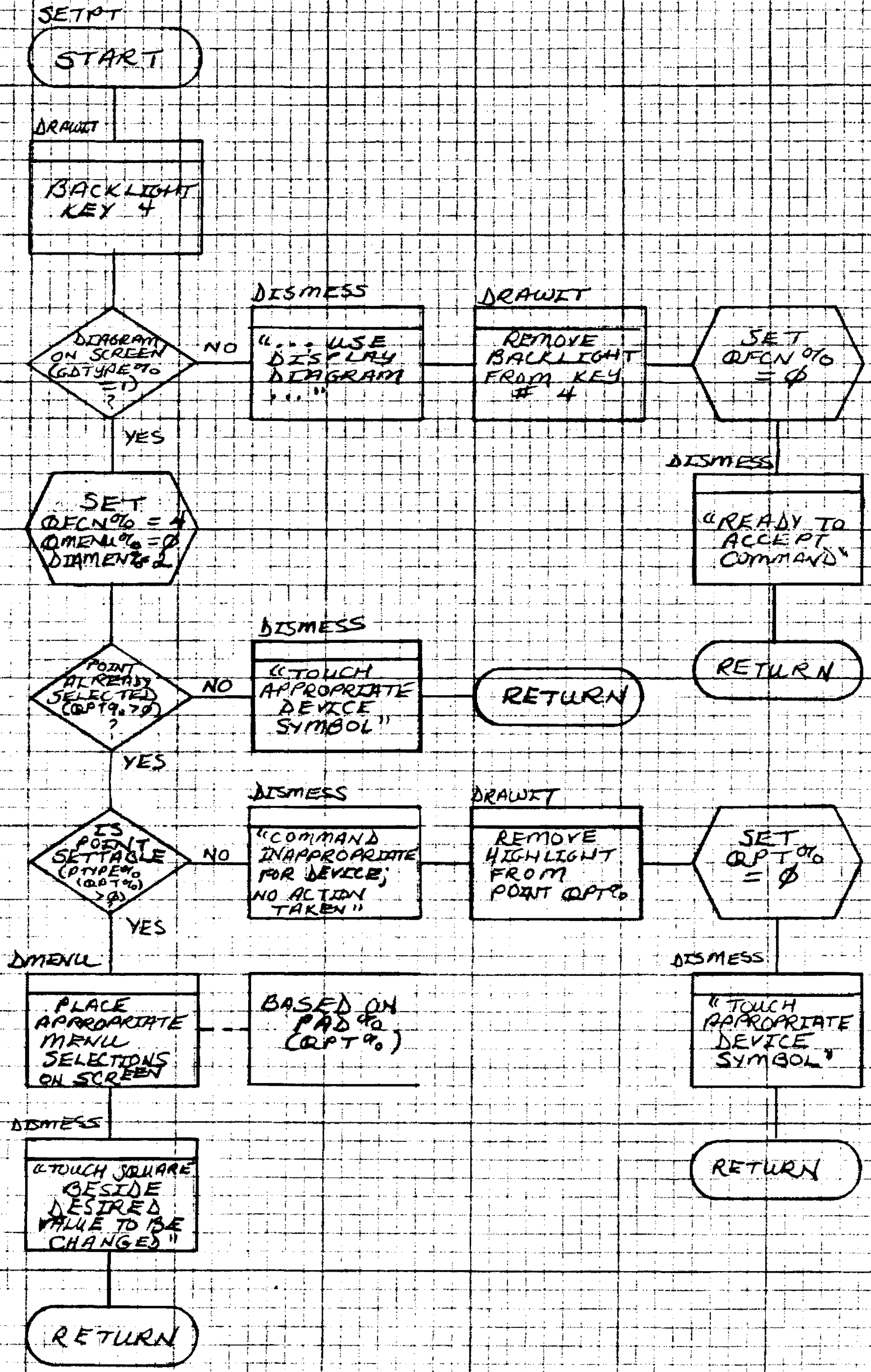
HARDWARE INTERACTION:

(none)

DESIGN NOTES:

(none)

SETPT: PROCESS "SET POINT/LIMITS" REQUEST p. 1 of 1



NAME: AUTO

PURPOSE:

The AUTO subroutine is used to process the AUTO/MANUAL command request. AUTO does not actually execute the command, but rather ensures that all conditions are right for its execution.

OPERATIONAL DESCRIPTION:

The AUTO/MANUAL key is backlighted and checks are made to ensure that a DE diagram is currently displayed and to determine whether or not a point has been selected. Based on these checks, the appropriate control and selection values variable are set and appropriate cues and error messages are given.

CALLED BY:

FKEYIN

CALLS:

DISMESS
DRAWIT

PASSED ARGUMENTS:

AMPT% - Auto/manual flag of point
DE% - Data environment auto/manual indicator
DIAMEN%- Number of diagram or menu currently displayed
GDTYPE%- Graphics display type
NDE$ - Textual name of data environment
PHIER% - Next higher point in DE hierarchy
PNAME$ - Textual name of point
QPT% - Selected point

NAME: AUTO (continued)

RETURNED ARGUMENT:

QAM% - Selected mode

QFCN% - Selected function

QPT% - Selected point

FILE INPUT/OUTPUT:

(none)

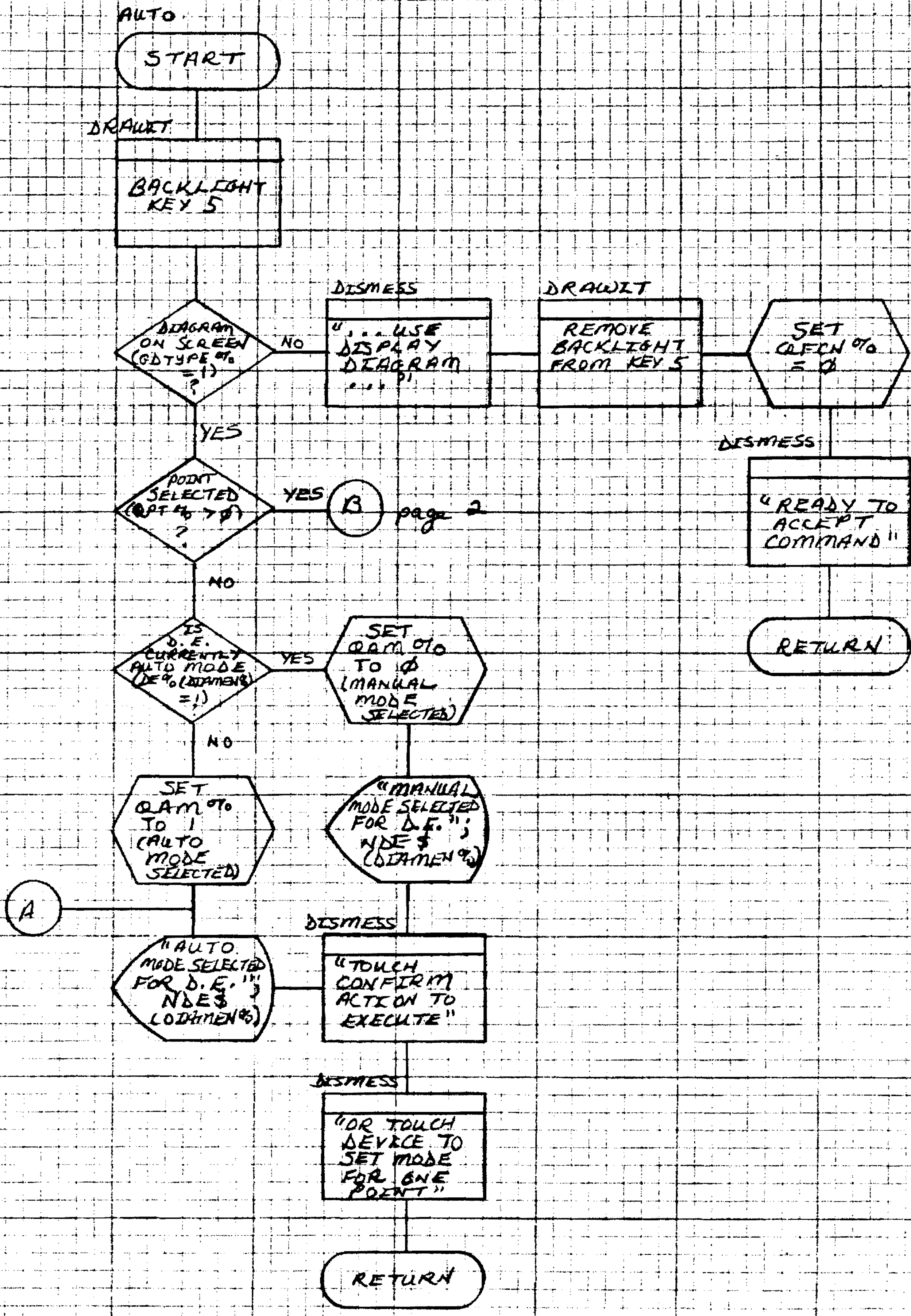
HARDWARE INTERACTION:

(none)

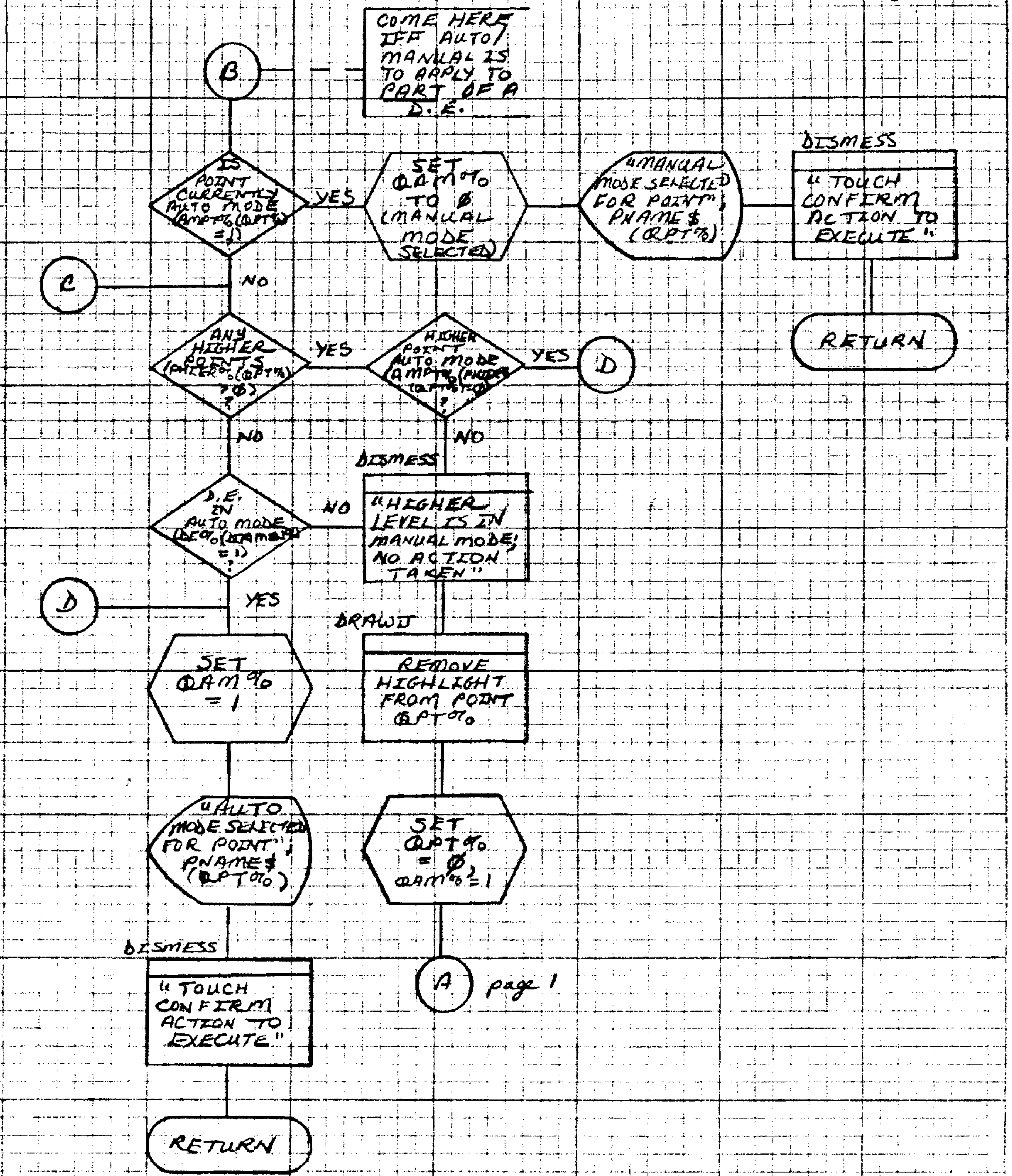
DESIGN NOTES:

(none)

AUTO: PROCESS "AUTO/MANUAL" REQUEST



AUTO: PROCESS "AUTO/MANUAL" REQUEST



NAME: REPORT

PURPOSE:

The REPORT subroutine is used to process the PRINT REPORT command request.

OPERATIONAL DESCRIPTION:

The PRINT REPORT function key is backlighted and the values for the control and selection variables are set. The DMENU subroutine is called to display the menu of available reports.

CALLED BY:

FKEYIN

CALLS:

DISMESS
DMENU
DRAWIT

PASSED ARGUMENTS:

(none)

RETURNED ARGUMENTS:

DIAMEN% - Number of diagram or menu currently displayed
QFCN% - Selected function
QMENU% - Selected menu item
QPT% - Selected point

NAME: REPORT (continued)

FILE INPUT/OUTPUT:

(none)

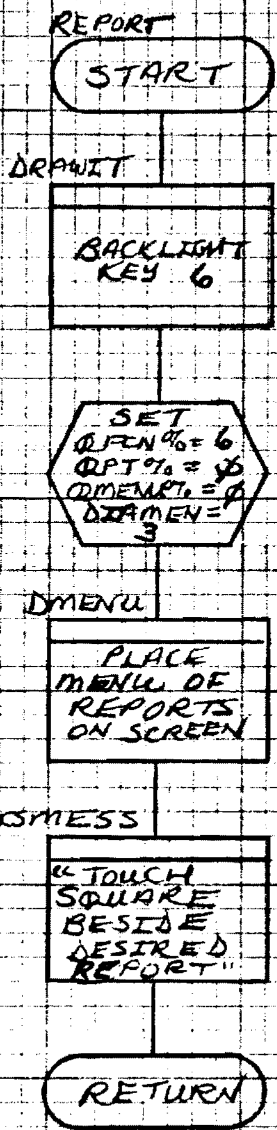
HARDWARE INTERACTION:

(none)

DESIGN NOTES:

(none)

REPORT: PROCESS "PRINT REPORT" REQUEST



NAME: SCHED

PURPOSE:

The SCHED subroutine is used to process the MODIFY SCHED command request.

OPERATIONAL DESCRIPTION:

The MODIFY SCHED function key is backlighted and a check is performed to ensure that a DE diagram is currently displayed. Selection and control variables are set and DMENU is called to display the menu of schedule modifications on the screen.

CALLED BY:

FKEYIN

CALLS:

DISMESS
DMENU
DRAWIT

PASSED ARGUMENTS:

GDTYPE% - Graphics display

RETURNED ARGUMENTS:

DIAMEN% - Number of diagram or menu currently displayed
QFCN% - Selected function
QMENU% - Selected menu item
QSCH% - Selected data environment schedule
S(QSCH%)- Current data environment schedule

NAME: SCHED (continued)

FILE INPUT/OUTPUT:

(none)

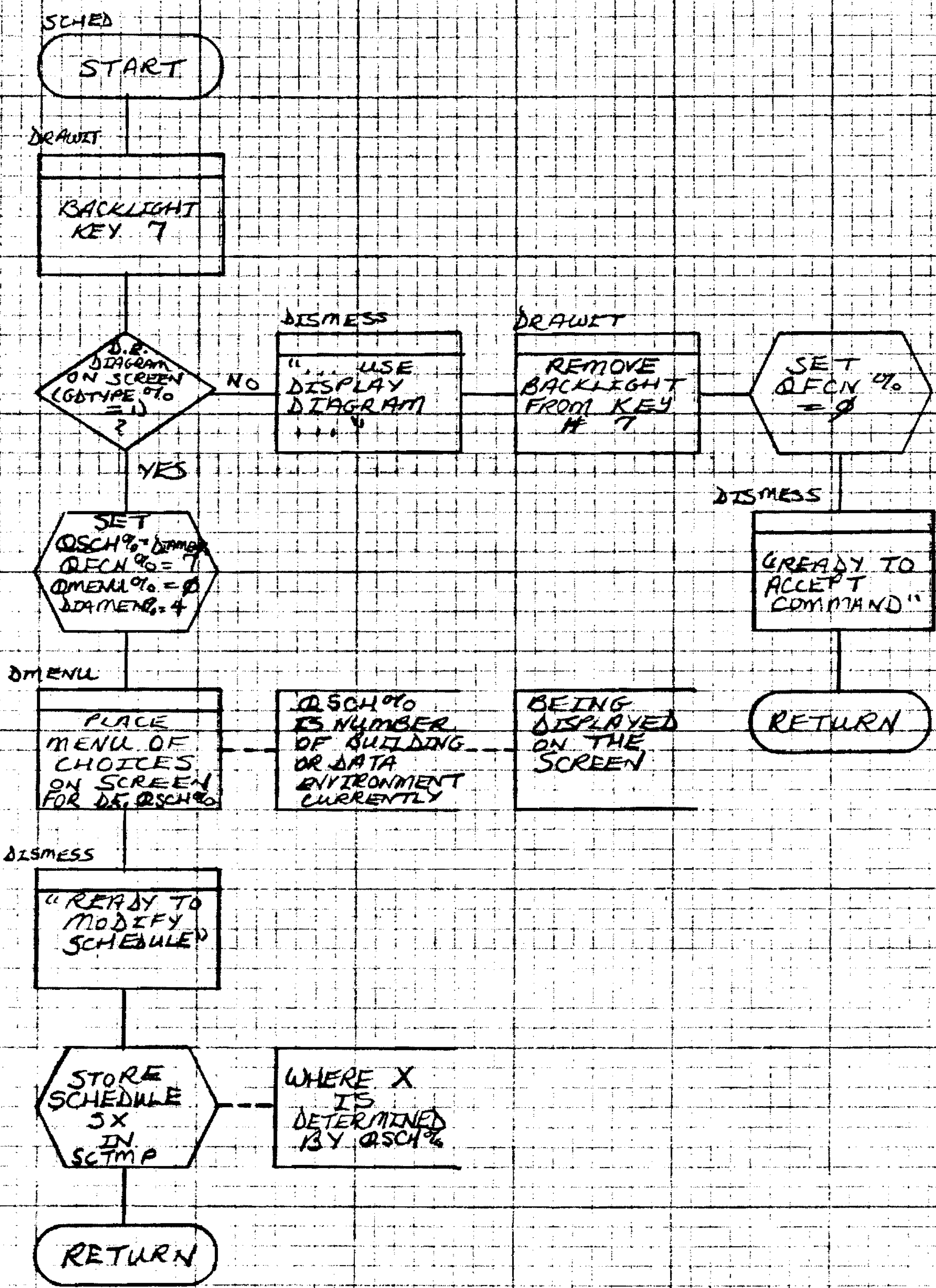
HARDWARE INTERACTION:

(none)

DESIGN NOTES:

(none)

SCHED: PROCESS "MODIFY SCHED" REQUEST p. 1 of 1



NAME: OPER

PURPOSE:

The OPER subroutine is used to process the CHANGE OPER request.

OPERATIONAL DESCRIPTION:

The CHANGE OPER function key is backlighted and the values for the control variables are set. Keyboard input is enabled and KEYIN is called to get the keyboard input string. If the string matches either the HELP or STOP commands, those subroutines are called. Otherwise, the input string is assumed to be the name of the new operator and the system prepares to change operators. KEYIN is used again to get the password. Both the operator's name and the password are verified against those stored in the system data base. If the name and password are correct, DRAWIT is called to change the DATE/TIME/OPERATOR display to show the new operator's name and the variable QOPER% is set to the index number for the new operator. Appropriate cue and error messages are given throughout the CHANGE OPER function. Touch Panel input is reenabled before program control returns to the calling routine.

CALLED BY:

FKEYIN

CALLS:

DISMESS
DRAWIT
HELP
KEYIN

NAME: OPER (continued)

PASSED ARGUMENTS:

OPER$ - String containing name of Ith operator

NOPER%- Number of system operators

PASS$ - String containing password for Ith operator

RETURNED ARGUMENTS:

KBTP% - Currently enabled interrupt

QAM% - Selected mode

QFCN% - Selected function

QMENU%- Selected menu item

QOPER%- Current operator

QPT% - Selected point

STOP% - Stop flag

FILE INPUT/OUTPUT:

(none)

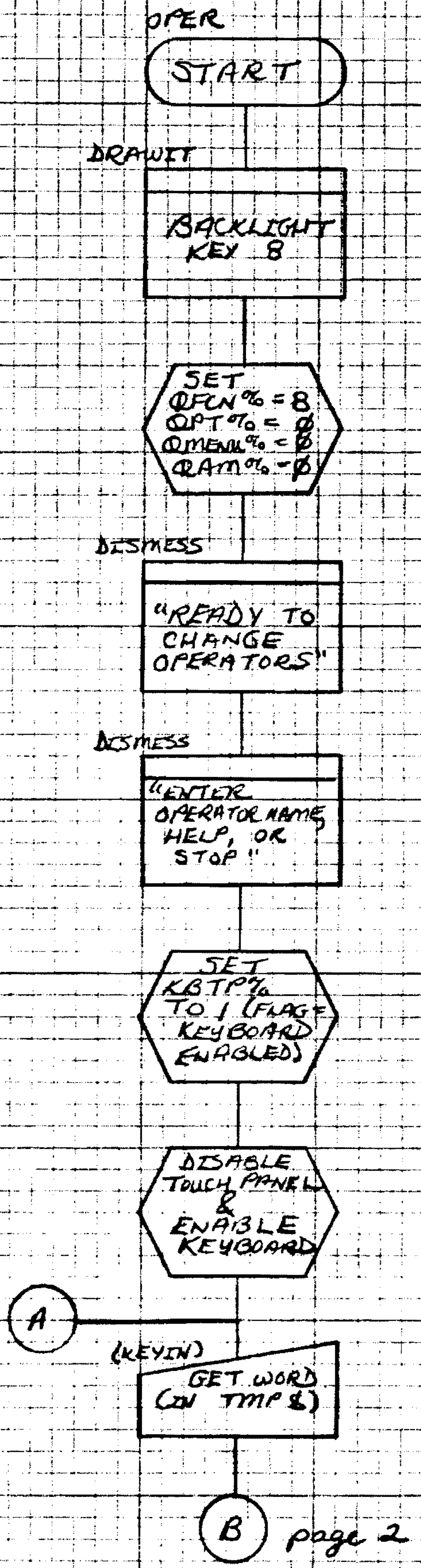
HARDWARE INTERACTION:

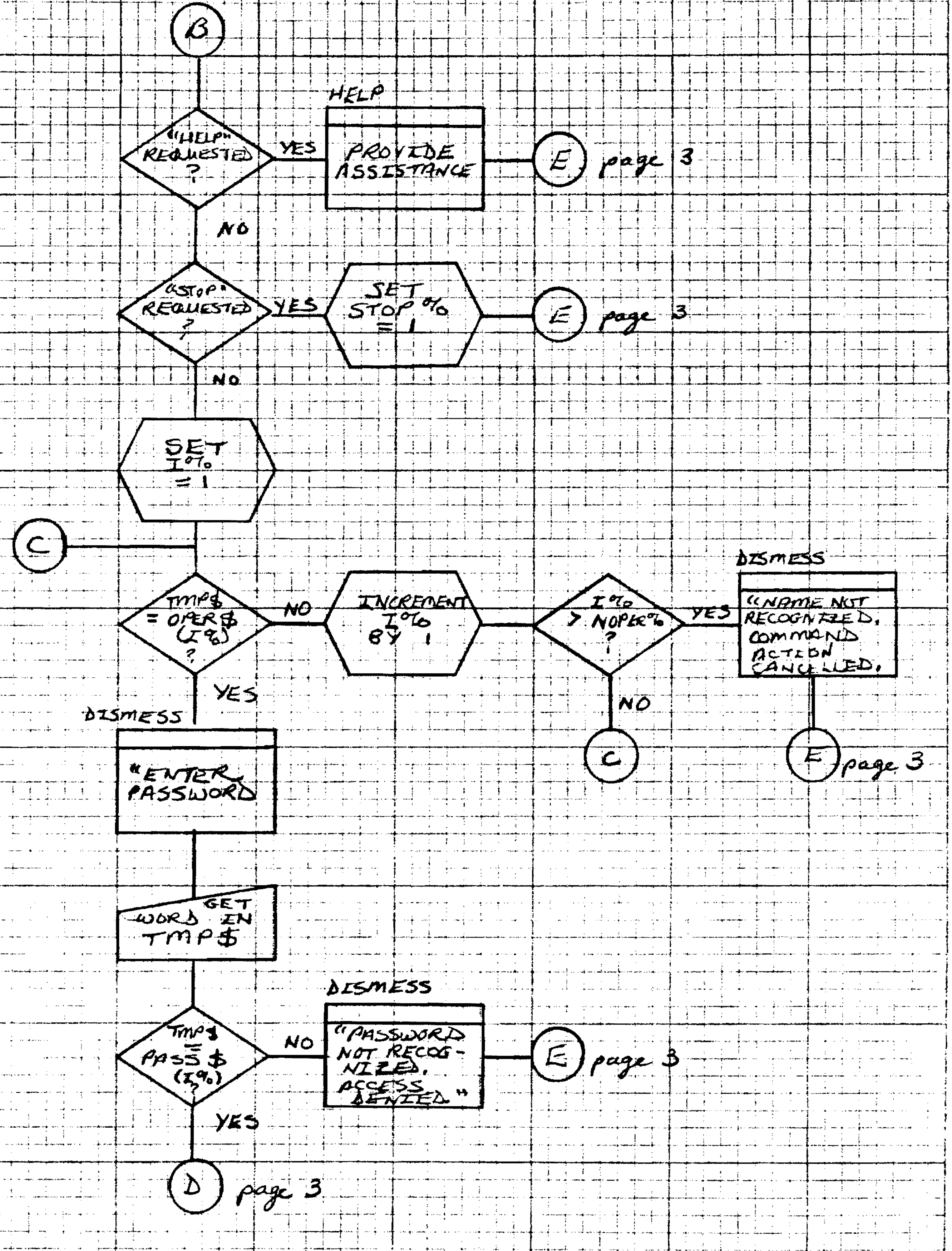
(none)

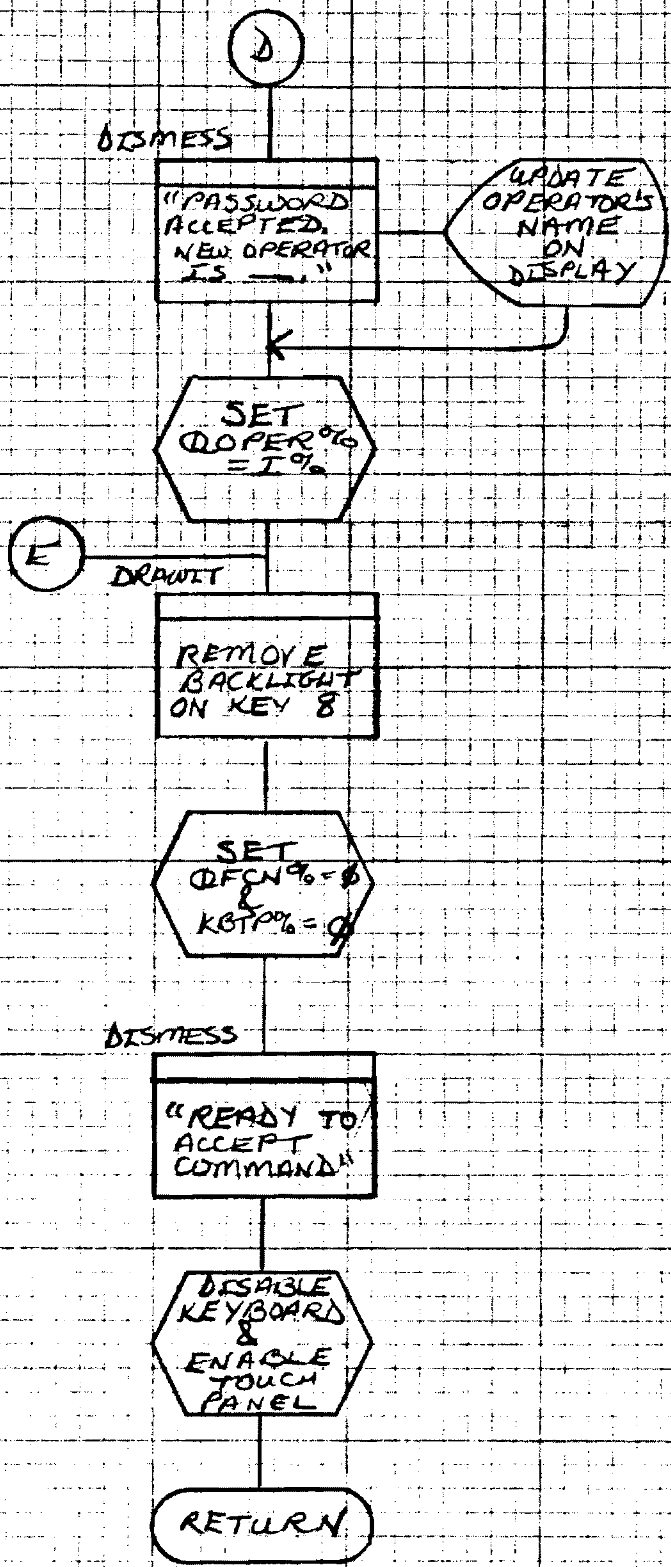
DESIGN NOTES:

(none)

OPER: PROCESS "CHANGE OPER" REQUEST




B
HELP
"HELP" REQUESTED ?
YES
PROVIDE ASSISTANCE
E
page 3
NO
"STOP" REQUESTED ?
YES
SET STOP% = 1
E
page 3
NO
SET I% = 1
C
TMP$ = OPER$ (I%) ?
NO
INCREMENT I% BY 1
I% > NOPER% ?
YES
DISMESS
"NAME NOT RECOGNIZED, COMMAND ACTION CANCELLED,
E
page 3
NO
C
YES
DISMESS
"ENTER PASSWORD
GET WORD IN TMP$
DISMESS
TMP$ = PASS$ (I%) ?
NO
"PASSWORD NOT RECOGNIZED. ACCESS DENIED"
E
page 3
YES
D
page 3


D
DISMESS
"PASSWORD ACCEPTED. NEW OPERATOR IS ___."
UPDATE OPERATOR'S NAME ON DISPLAY
SET ØOPER% = I%
E
DRAWIT
REMOVE BACKLIGHT ON KEY 8
SET ØFCN% = Ø & KBTP% = Ø
DISMESS
"READY TO ACCEPT COMMAND"
DISABLE KEYBOARD & ENABLE TOUCH PANEL
RETURN

NAME: CONFIRM

PURPOSE:

The CONFIRM subroutine is used to process the CONFIRM ACTION command request.

OPERATIONAL DESCRIPTION:

The CONFIRM ACTION key is backlighted and checks are made to ensure that a function and all its parameters have been selected. Appropriate error messages are given for improper or incomplete selections. If all selections have been made, then based upon which function has been selected, the appropriate subroutine is called to execute the function.

CALLED BY:

FKEYIN

CALLS:

DISMESS
DRAWIT
XAM
XDD
XMS
XPR
XSD
XSE
XSPL

NAME: CONFIRM (continued)

PASSED ARGUMENTS:

NDES%	- Number of data environments in system
NREPORTS%	- Number of available reports
QAM%	- Selected mode
QFCN%	- Selected function
QMENU%	- Selected menu item
QPT%	- Selected point

RETURNED ARGUMENTS:

QFCN% - Selected function

FILE INPUT/OUTPUT:

(none)

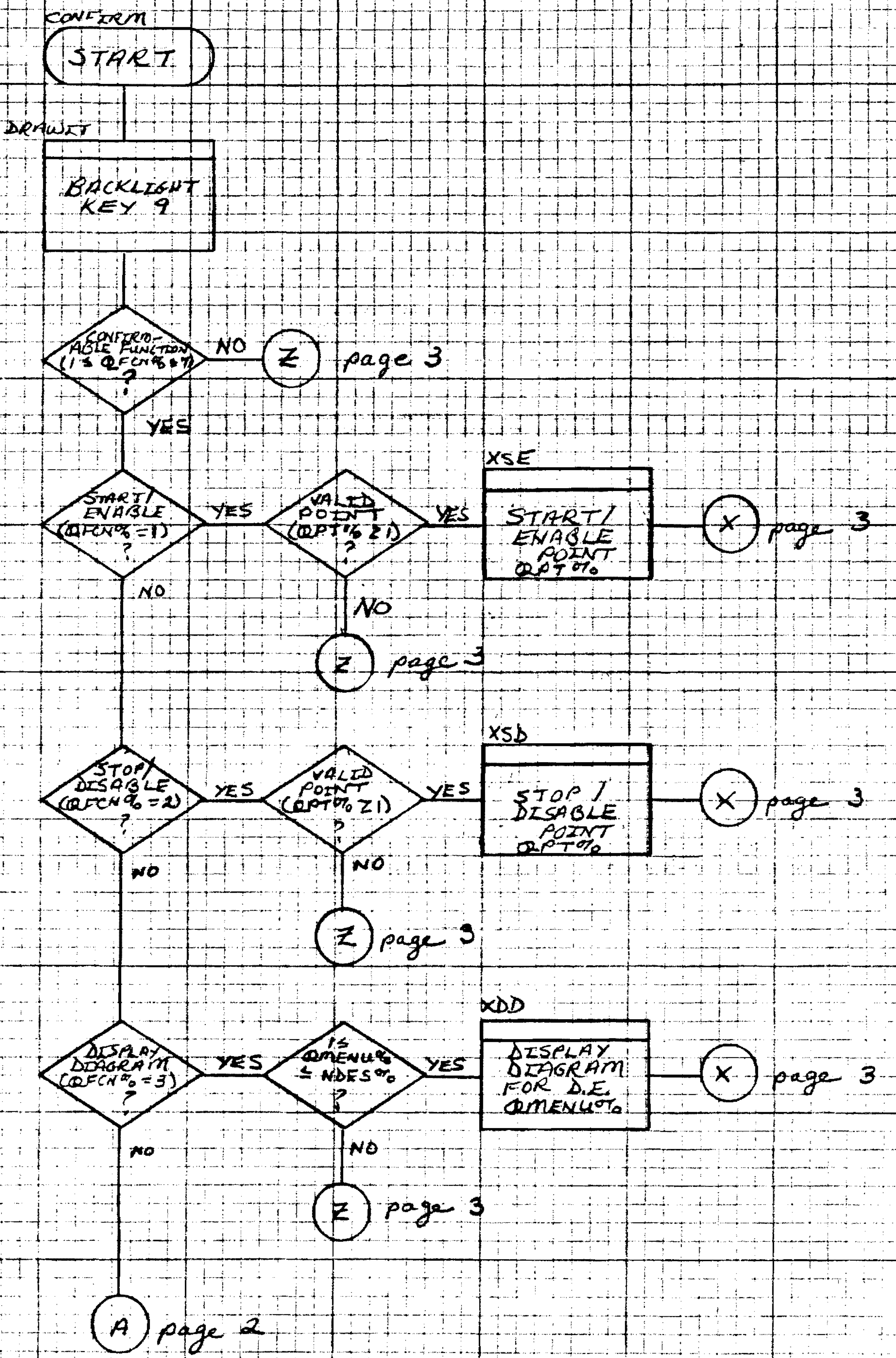
HARDWARE INTERACTION:

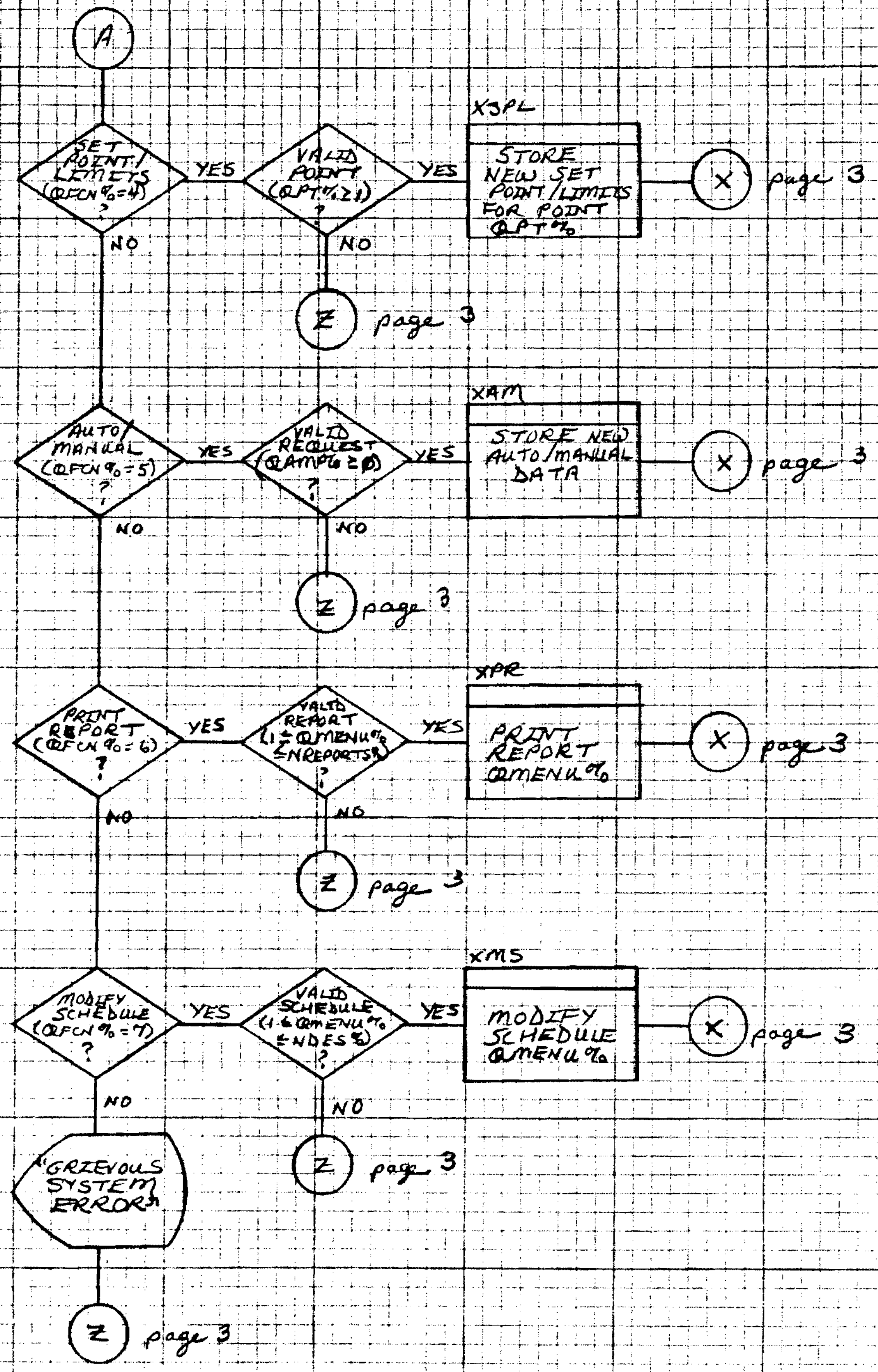
CRT DISPLAY - Print messages

DESIGN NOTES:

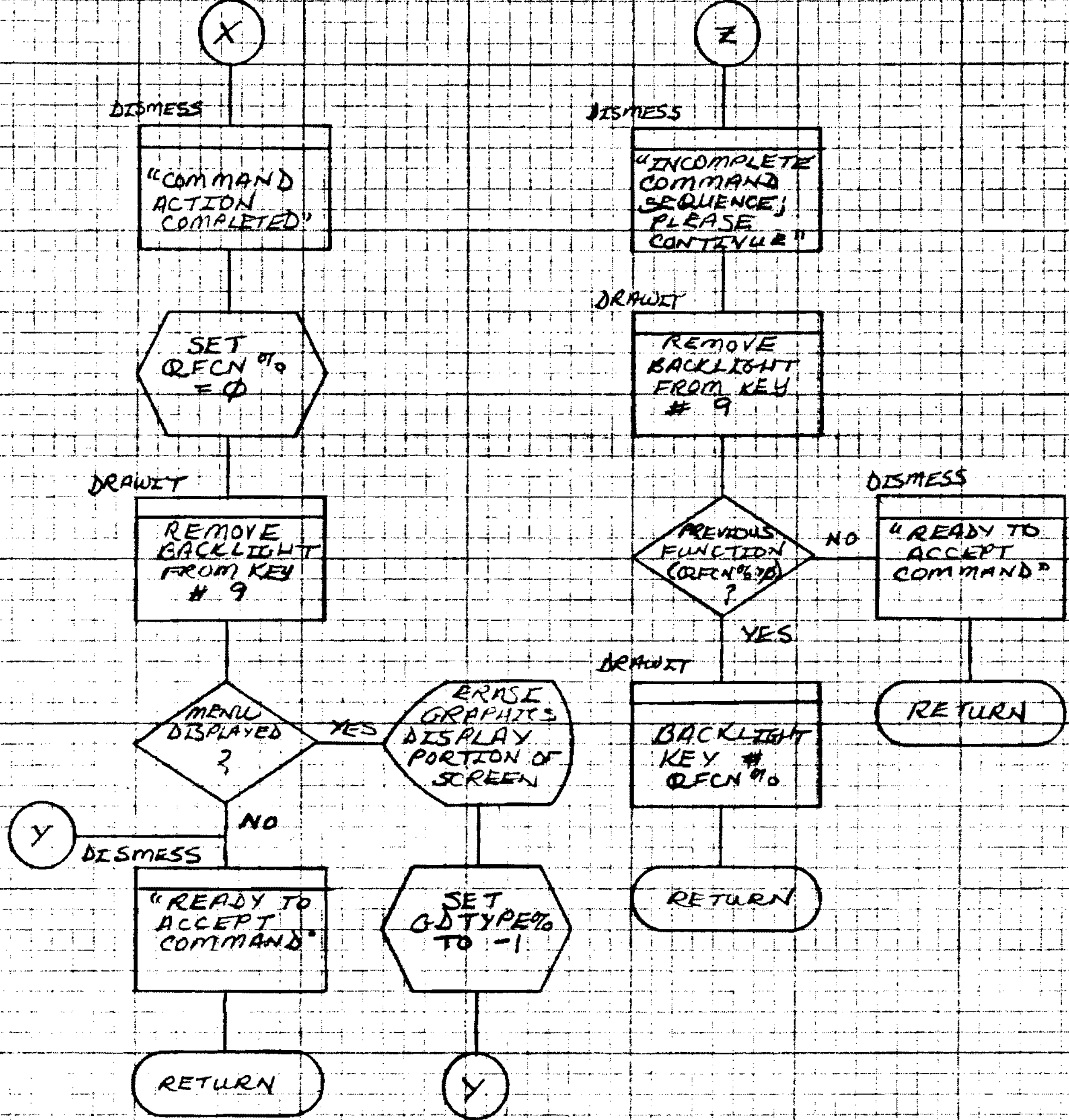
(none)

CONFIRM: PROCESS "CONFIRM ACTION" REQUEST




A
SET POINT/LIMITS (@FCN%=4) ?
YES
VALID POINT (@PT%≥1) ?
YES
XSPL
STORE NEW SET POINT/LIMITS FOR POINT @PT%
X
page 3
NO
Z
page 3
NO
AUTO/MANUAL (@FCN%=5) ?
YES
VALID REQUEST (@AMP%≥0) ?
YES
XAM
STORE NEW AUTO/MANUAL DATA
X
page 3
NO
Z
page 3
NO
PRINT REPORT (@FCN%=6) ?
YES
VALID REPORT (1≤@MENU% ≤NREPORTS%) ?
YES
XPR
PRINT REPORT @MENU%
X
page 3
NO
Z
page 3
NO
MODIFY SCHEDULE (@FCN%=7) ?
YES
VALID SCHEDULE (1≤@MENU% ≤NDES%) ?
YES
XMS
MODIFY SCHEDULE @MENU%
X
page 3
NO
Z
page 3
NO
"GRIEVOUS SYSTEM ERROR"
Z
page 3

CONFIRM: PROCESS "CONFIRM ACTION" REQUEST



NAME: CANCEL

PURPOSE:

Process the CANCEL ACTION command request.

OPERATIONAL DESCRIPTION:

The CANCEL ACTION key is backlighted and the message "Command Action Cancelled" is displayed. Highlighting is removed from the selected point, if any, and all selection variables are turned off. The function key backlight is removed and program control returns to the calling routine.

CALLED BY:

FKEYIN

CALLS:

DISMESS
DRAWIT

PASSED ARGUMENTS:

GDTYPE% - Graphics display type
QPT% - Selected point

RETURNED ARGUMENTS:

GDTYPE% - Graphics display type
QAM% - Selected mode
QFCN% - Selected function
QMENU% - Selected menu item
QPT% - Selected point

NAME: CANCEL (continued)

FILE INPUT/OUTPUT:

(none)

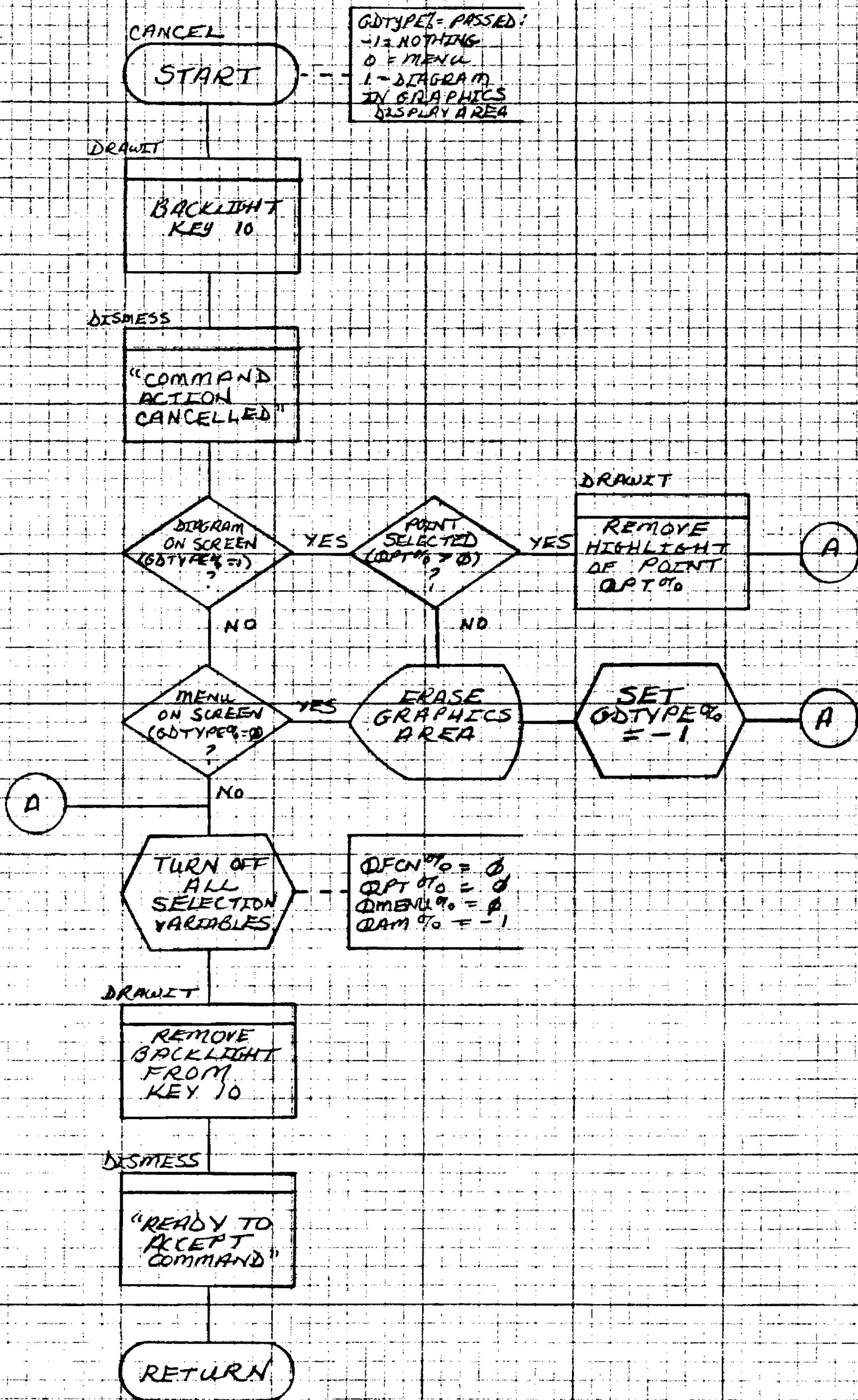
HARDWARE INTERACTION:

CRT DISPLAY - Manipulation of graphics area

DESIGN NOTES:

(none)

CANCEL: PROCESS "CANCEL ACTION" REQUEST page 1 of 1



NAME: MDIN

PURPOSE:

The MDIN subroutine is used to process the menu of diagrams selection.

OPERATIONAL DESCRIPTION:

The MDIN subroutine uses the coordinates of the touch panel input point to determine which menu item was selected. DRAWIT is then called to highlight the selected item and the menu selection variable is set.

CALLED BY:

MENUIN

CALLS:

DISMESS
DRAWIT

PASSED ARGUMENTS:

XTOUCH% - X-coordinate of touched point
YTOUCH% - Y-coordinate of touched point

RETURNED ARGUMENTS:

QMENU% - Selected menu item

FILE INPUT/OUTPUT:

(none)

NAME: MDIN (continued)

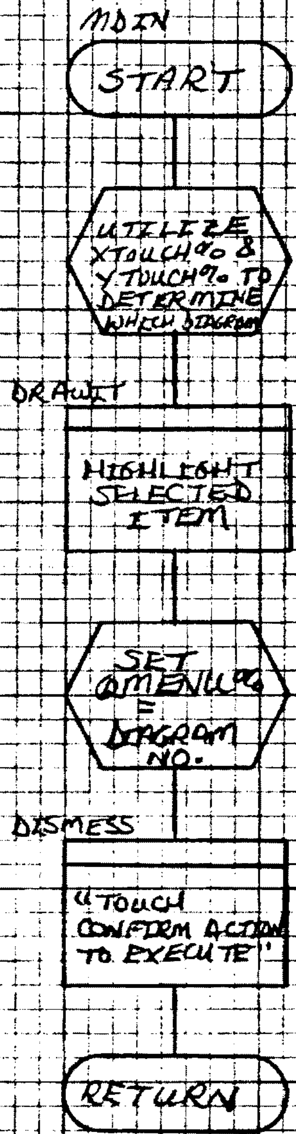
HARDWARE INTERACTION:

(none)

DESIGN NOTES:

(none)

MDIN: PROCESS MENU OF DIAGRAMS



NAME: MLIN

PURPOSE:

The MLIN subroutine is used to process the menu of SET POINTS/LIMITS selection request.

OPERATIONAL DESCRIPTION:

The coordinates of the touch panel input point are used to determine which item was selected. DRAWIT is called to highlight the selection, and keyboard input is enabled. When the new value is read from the keyboard it is checked to make sure that it is valid. An invalid value will result in an error message; otherwise, the value is stored in temporary storage. DRAWIT is then used to remove the highlight from the selection, and touch panel input is reenabled.

CALLED BY:

MENUIN

CALLS:

DISMESS

MRAWIT

PASSED ARGUMENTS:

HILIM - High limit for analog point

LOWLIM - Low limit for analog point

XTOUCH% - X-coordinate of touched point

YTOUCH% - Y-coordinate of touched point

NAME: MLIN (continued)

RETURNED ARGUMENTS:

- ATMP - Temporary storage for digital alarm value
- HTMP - Temporary storage for analog high limit
- KBTP% - Currently enabled interrupt
- LTMP - Temporary storage for analog low limit
- QMENU% - Selected menu item
- STMP - Temporary storage for analog set point

FILE INPUT/OUTPUT:

(none)

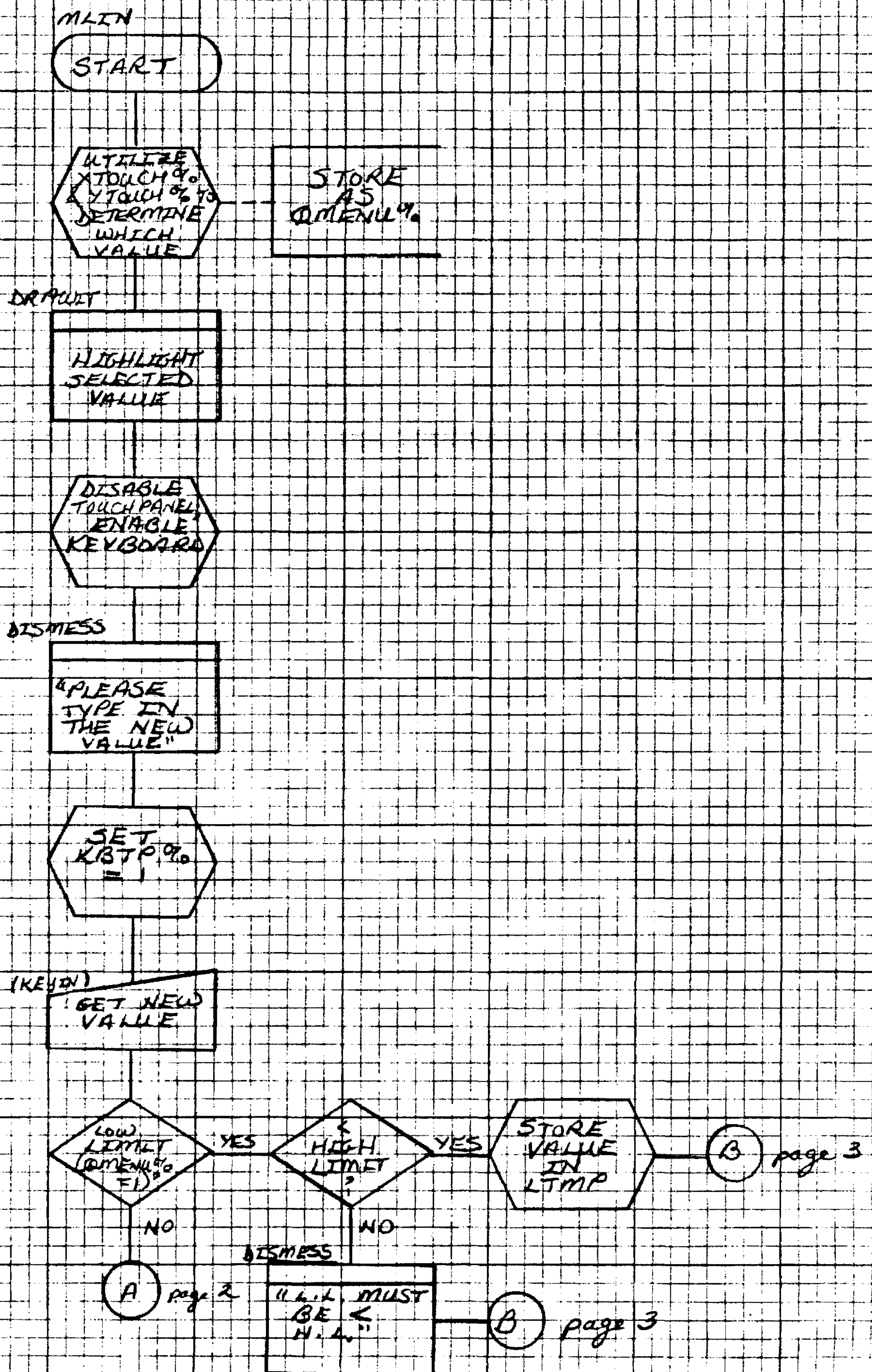
HARDWARE INTERACTION:

(none)

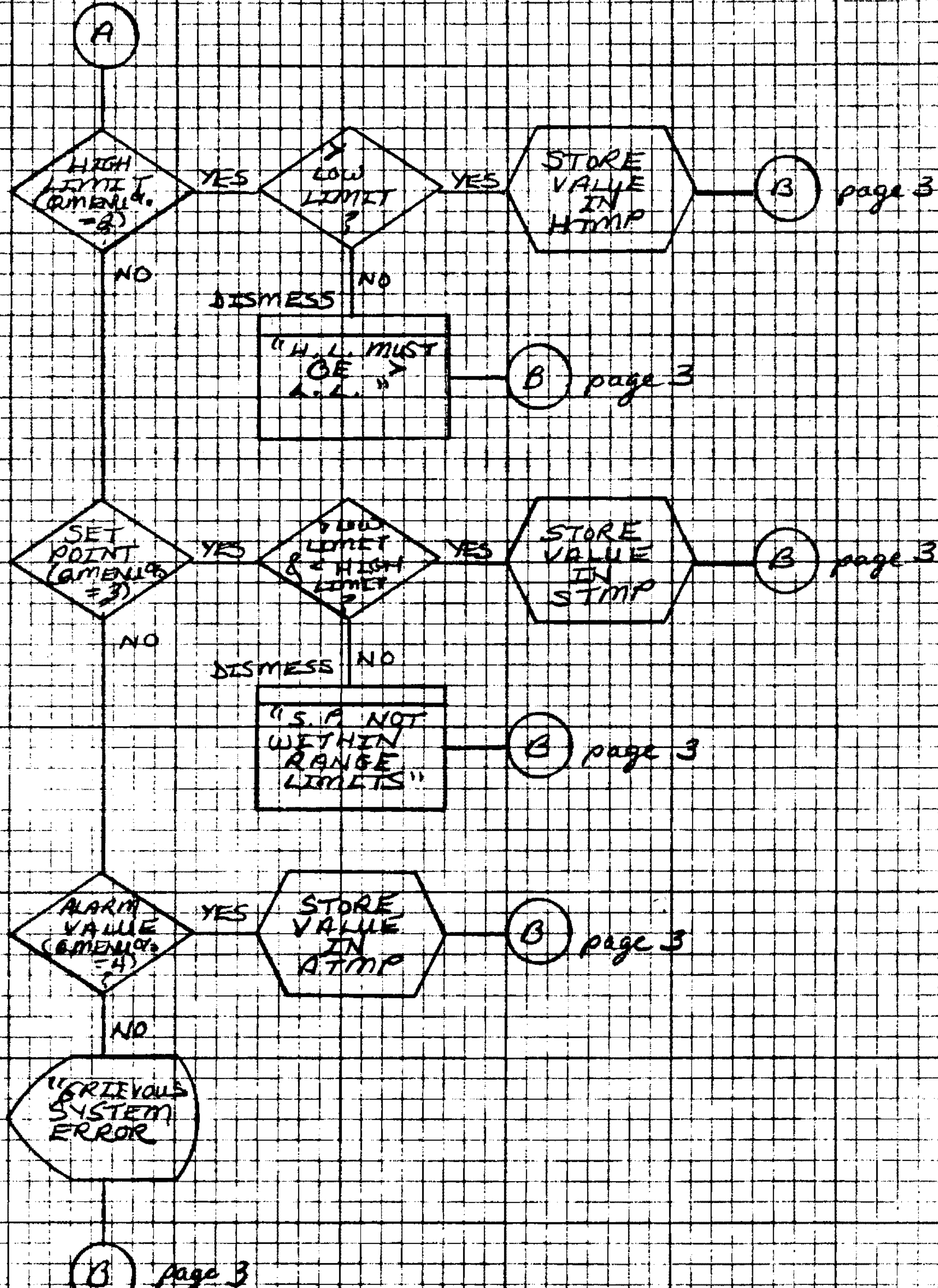
DESIGN NOTES:

(none)

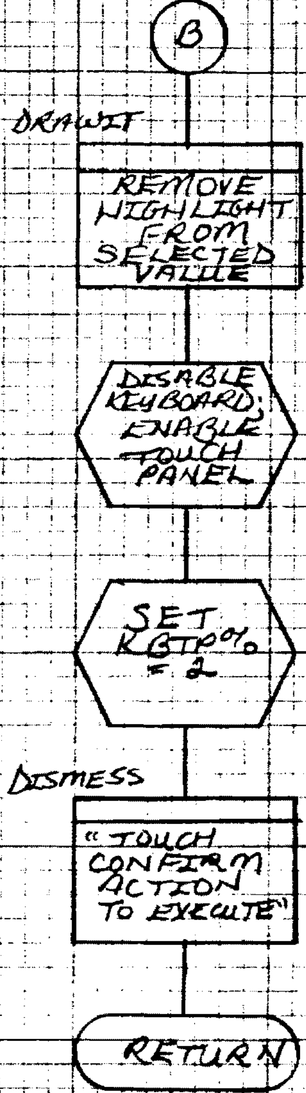
MLIN: PROCESS MENU OF SET POINTS/LIMITS



MLIN: PROCESS MENU OF SET POINTS/LIMITS



MLIN: PROCESS MENU OF SET POINTS/LIMITS



NAME: MRIN

PURPOSE:

The MRIN subroutine is used to process the menu of reports selection.

OPERATIONAL DESCRIPTION:

The coordinates of the touch panel input point are used to determine which menu item has been selected. DRAWIT is called to highlight the selection, and the selection control variable is set.

CALLED BY:

MENUIN

CALLS:

DISMESS
DRAWIT

PASSED ARGUMENTS:

XTOUCH% - X-coordinate of touched point
YTOUCH% - Y-coordinate of touched point

RETURNED ARGUMENTS:

QMENU% - Selected menu item.

FILE INPUT/OUTPUT:

(none)

NAME: MRIN (continued)

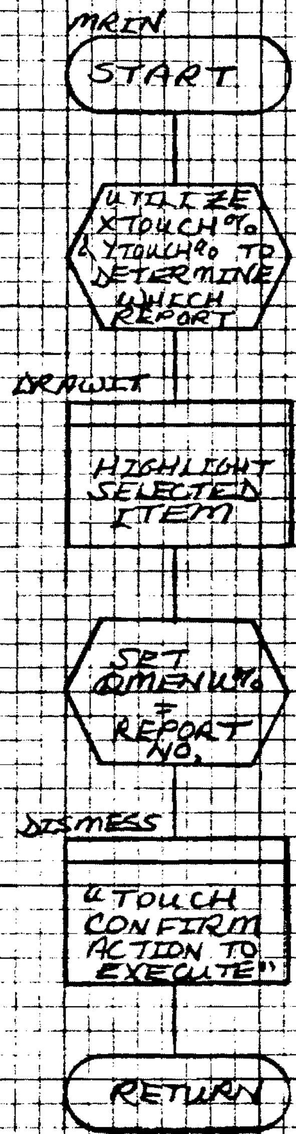
HARDWARE INTERACTION:

(none)

DESIGN NOTES:

(none)

MRIN: PROCESS MENU OF REPORTS



NAME: MSIN

PURPOSE:

The MSIN subroutine is used to process the menu of schedules selection.

OPERATIONAL DESCRIPTION:

The coordinates of the touch panel input point are used to determine which menu item was selected. DRAWIT is used to highlight the selection, and keyboard input is enabled. When the new schedule is read from the keyboard, it is checked for validity. An invalid value will result in an error message; othewise, the new schedule is stored in temporary storage. DRAWIT is used to remove the highlight from the selection, and touch panel input is reenabled.

CALLED BY:

MENUIN

CALLS:

DISMESS
DRAWIT

PASSED ARGUMENTS:

X-TOUCH% - X-coordinate of touched point
Y-TOUCH% - Y-coordinate of touched point

NAME: MSIN (continued)

RETURNED ARGUMENTS:

KBTP% - Currently enabled interrupt

QMENU% - Selected menu item

SCTMP - Temporary storage for new (modified) schedule

FILE INPUT/OUTPUT:

(none)

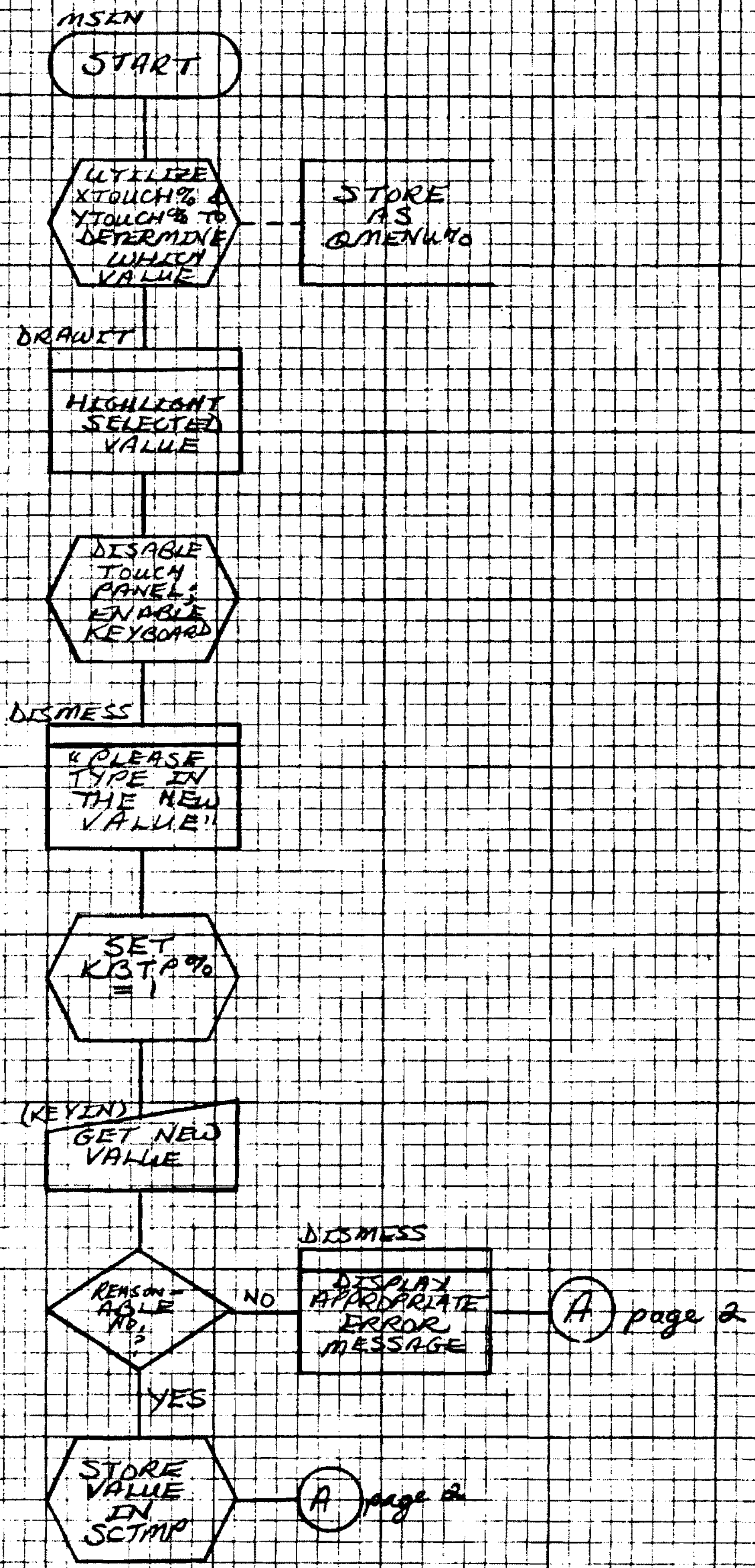
HARDWARE INTERACTION:

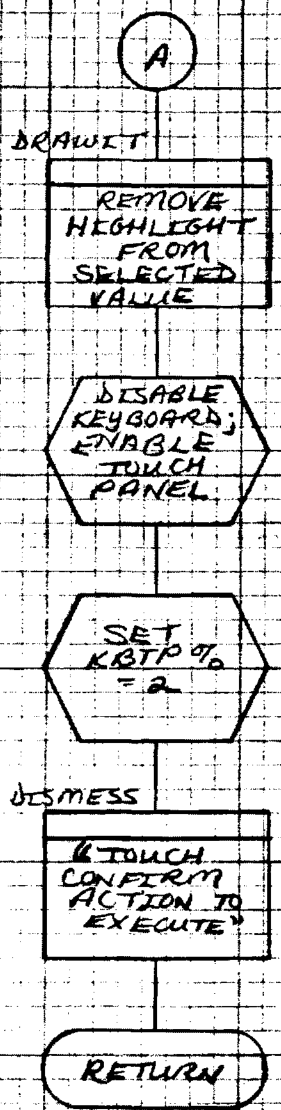
(none)

DESIGN NOTES:

(none)

MSCN: PROCESS "MENU" OF SCHEDULES




A
DRAWIT
REMOVE HIGHLIGHT FROM SELECTED VALUE
DISABLE KEYBOARD; ENABLE TOUCH PANEL
SET KBTP% = 2
DISMESS
"TOUCH CONFIRM ACTION TO EXECUTE"
RETURN

NAME: XSE

PURPOSE:

The XSE subroutine is used to execute the START/ENABLE function.

OPERATIONAL DESCRIPTION:

The XSE subroutine sets the status indicator for the selected point and all points tied to and below it in the DE process hierarchy to the started/enabled condition. It also readjusts the symbol coloration for the affected devices if they are currently displayed on the screen. Messages are given for each device which is affected.

CALLED BY:

CONFIRM

CALLS:

DRAWIT

PASSED ARGUMENTS:

DIAMEN% - Number of diagram or menu currently displayed
GDTYPE% - Graphics display type
LPOINT% - Number of diagram on which point is located
NRTPTS% - Number of real-time points
PTYPE% - Type of point
QPT% - Selected point

NAME: XSE (continued)

RETURNED ARGUMENTS:

DSTAT% - Device status

QPT% - Selected point

FILE INPUT/OUTPUT:

(none)

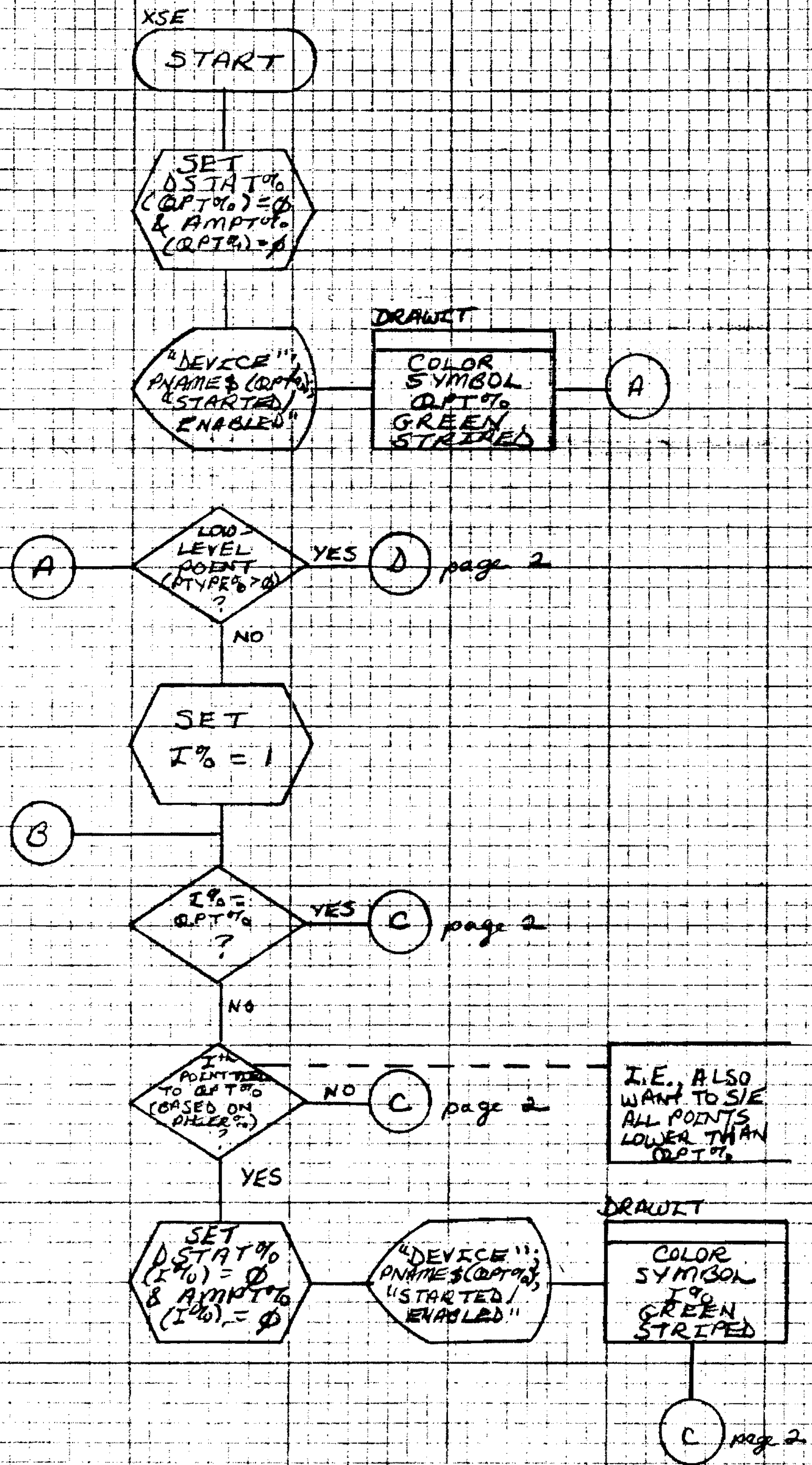
HARDWARE INTERACTION:

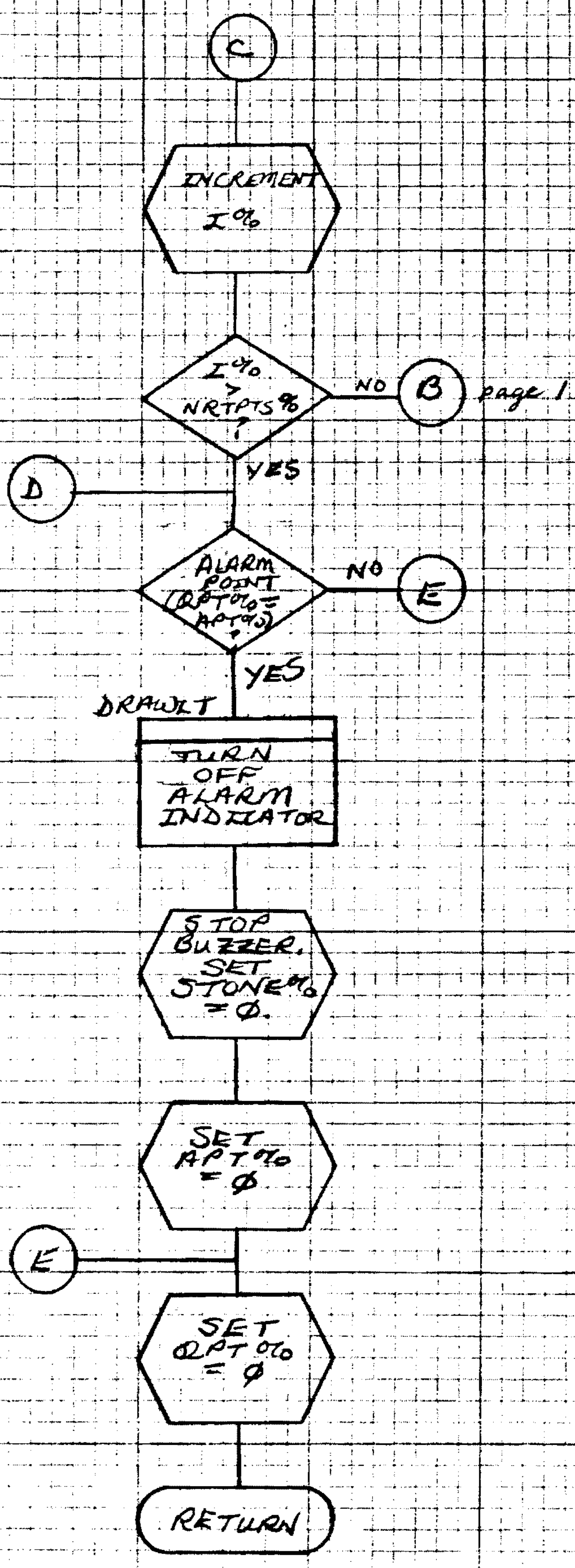
(none)

DESIGN NOTES:

(none)

XSE: EXECUTE START/ENABLE




C
INCREMENT I%
I% > NRTPTS% ?
NO
B
page 1
YES
D
ALARM POINT (QPT% = APT%) ?
NO
E
YES
DRAWIT
TURN OFF ALARM INDICATOR
STOP BUZZER. SET STONE% = Ø.
SET APT% = Ø
E
SET QPT% = Ø
RETURN

NAME: XSD

PURPOSE:

The XSD subroutine is used to execute the STOP/DISABLE function.

OPERATIONAL DESCRIPTION:

The XSD subroutine sets the status indicator for the selected point and all points tied to and below it in the DE process hierarchy to the stopped/disabled condition. It also readjusts the symbol coloration for the affected devices if they are currently displayed on the screen. Messages are given for each device which is affected.

CALLED BY:

CONFIRM

CALLS:

DRAWIT

PASSED ARGUMENTS:

DIAMEN% - Number of diagram or menu currently displayed
GDTYPE% - Graphics display type
LPOINT% - Number of diagram on which point is located
NRTPTS% - Number of real-time points
PTYPE% - Type of point
QPT% - Selected point

NAME: XSD

RETURNED ARGUMENTS:

DSTAT% - Device status

QPT% - Selected point

FILE INPUT/OUTPUT:

(none)

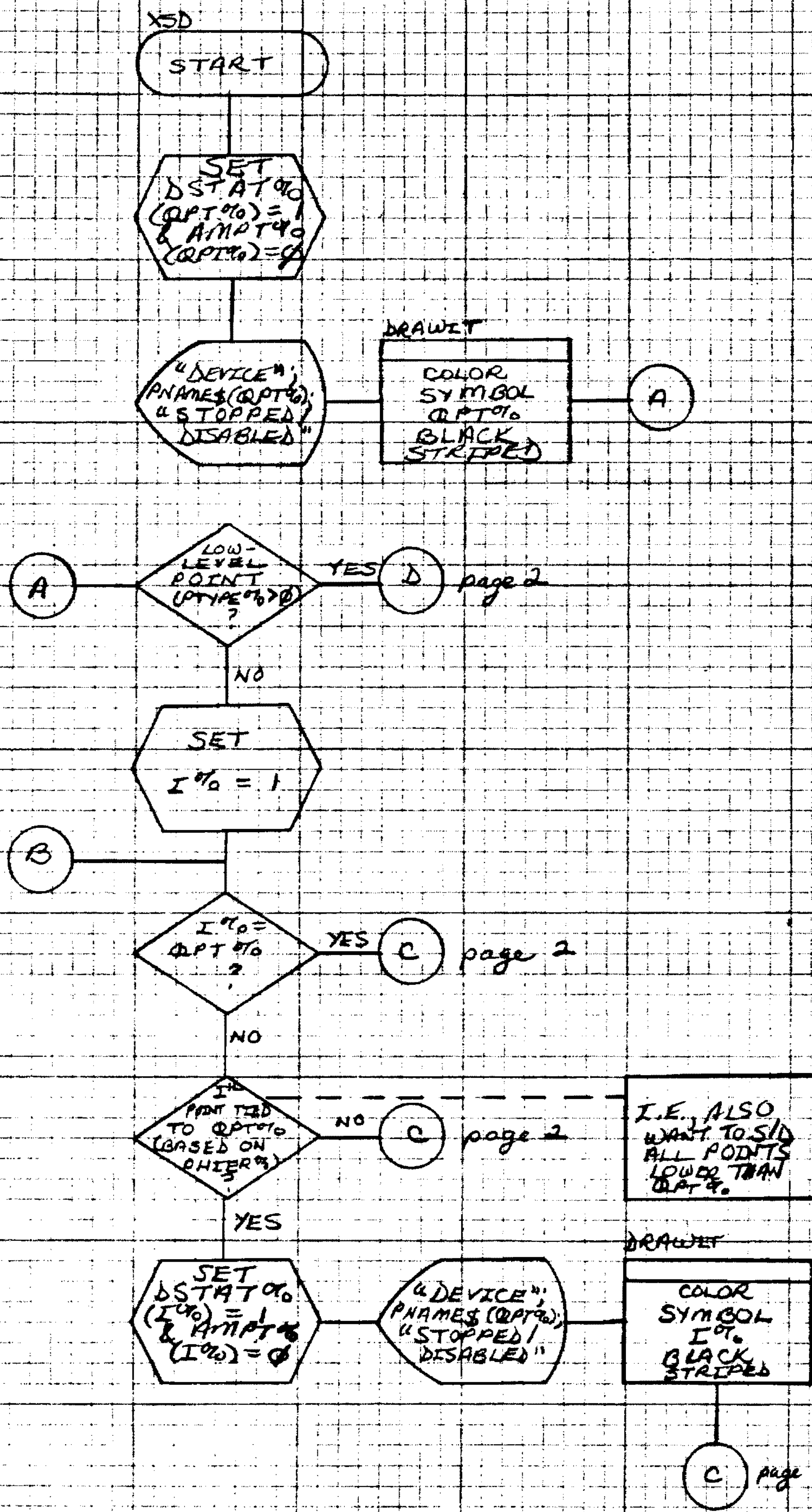
HARDWARE INTERACTION:

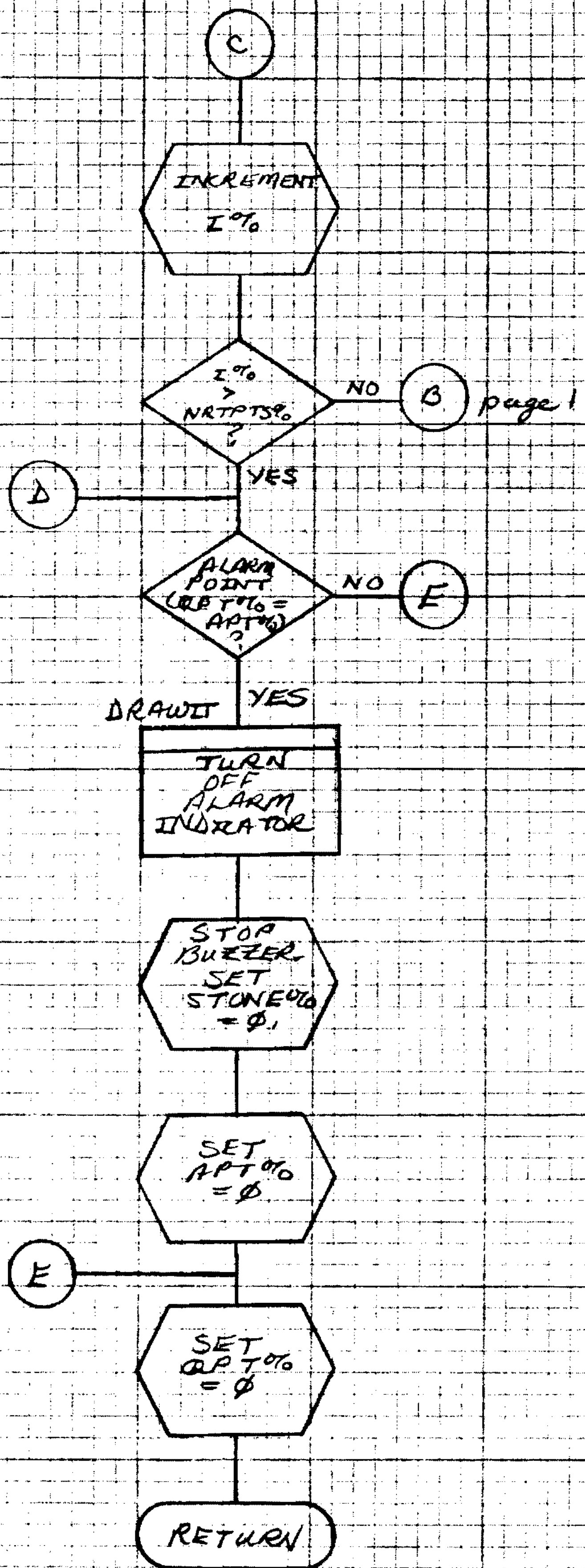
(none)

DESIGN NOTES:

(none)

XSD: EXECUTE STOP/DISABLE




C
INKREMENT I%
I% > NRTPTS% ?
NO
B
page 1
YES
D
ALARM POINT (RPT% = APT%) ?
NO
E
YES
DRAWIT
TURN OFF ALARM INDICATOR
STOP BUZZER, SET STONE% = Ø,
SET APT% = Ø
E
SET RPT% = Ø
RETURN

NAME: XDD

PURPOSE:

The XDD subroutine is used to execute the DISPLAY DIAGRAM command function.

OPERATIONAL DESCRIPTION:

A filename is constructed which corresponds to a disk file containing the buffer memory image of the selected data environment diagram. This buffer file contains the interconnecting lines for the symbols to be displayed. This file is loaded into memory and is displayed on the screen. DRAWIT is then used to place the correct symbols on the diagram using appropriate colors, rotation, etc. Colors and flashing mode are selected on the basis of the values stored for each point in the real-time data base.

CALLED BY:

CONFIRM

CALLS:

DRAWIT

PASSED ARGUMENTS:

AMPT% - Auto/Manual indicator for Ith point
APT% - Current alarm point
ASTAT% - Alarm status indicator for Ith point
DBVAL - Real-time data base value for Ith point
DE% - Auto/manual indicator for DE
DIAMEN%- Number of diagram or menu currently displayed
DSTAT% - Disable status flag for Ith point

NAME: XDD (continued)

PASSED ARGUMENTS: (concluded)

LPOINT%- Diagram on which Ith point is located

NRTPTS%- Number of real-time points

QMENU% - Selected item

RETURNED ARGUMENTS:

DIAMEN%- Number of diagram or menu currently displayed

GDTYPE%- Graphics display type

QMENU% - Selected menu item

FILE INPUT/OUTPUT:

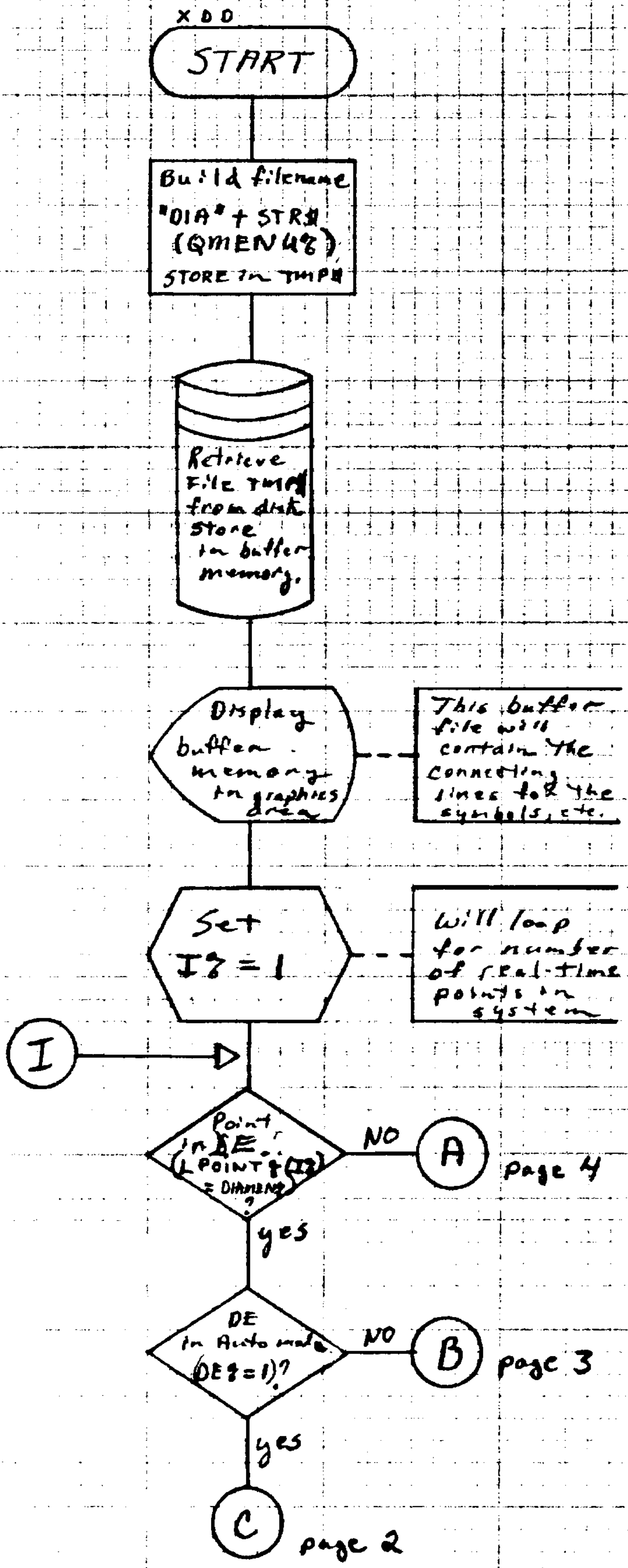
Files are retrieved from disk and loaded into memory for subsequent display on the screen. Filenames are of the form DIA*, where * is the number of the data environment to be displayed.

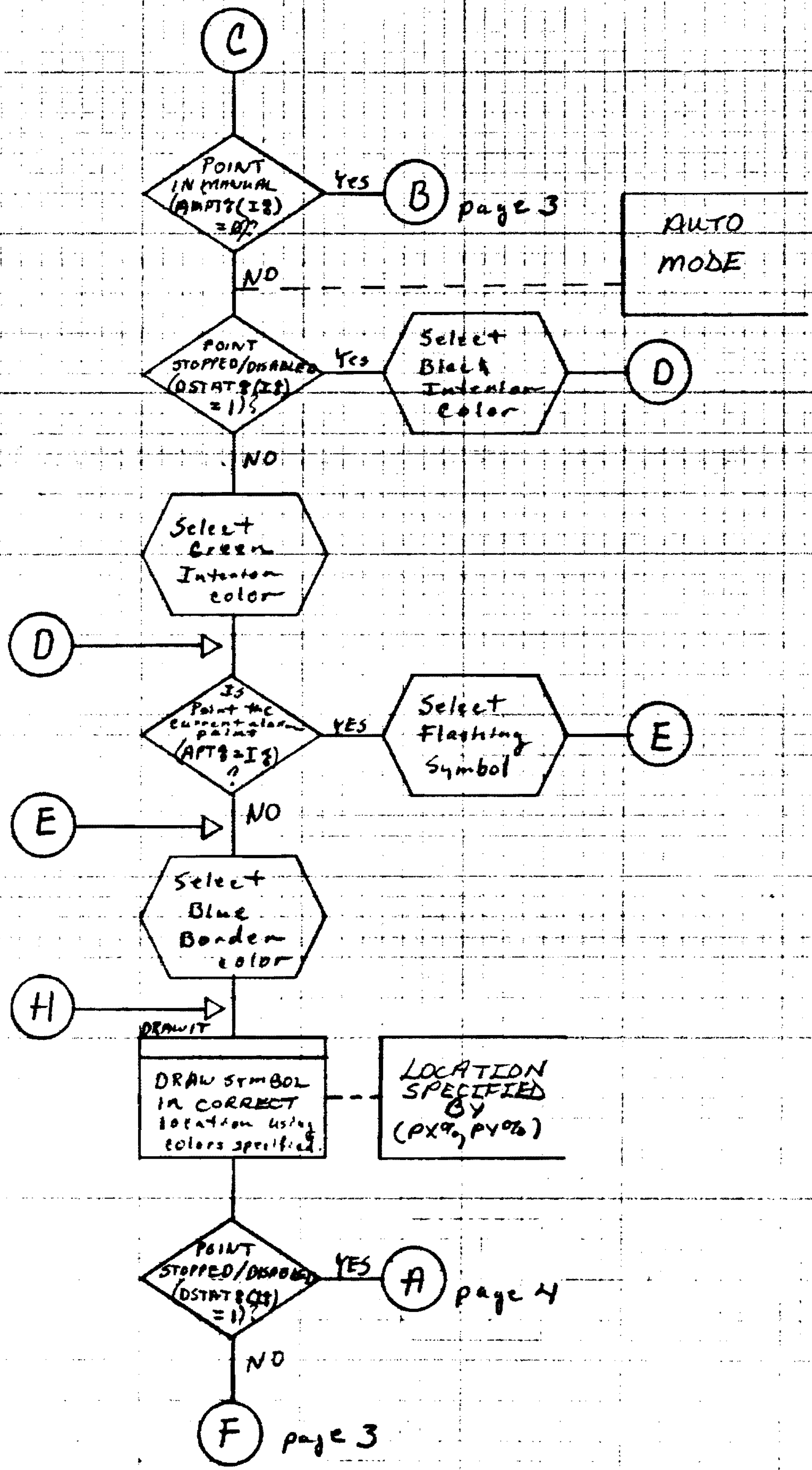
HARDWARE INTERACTION:

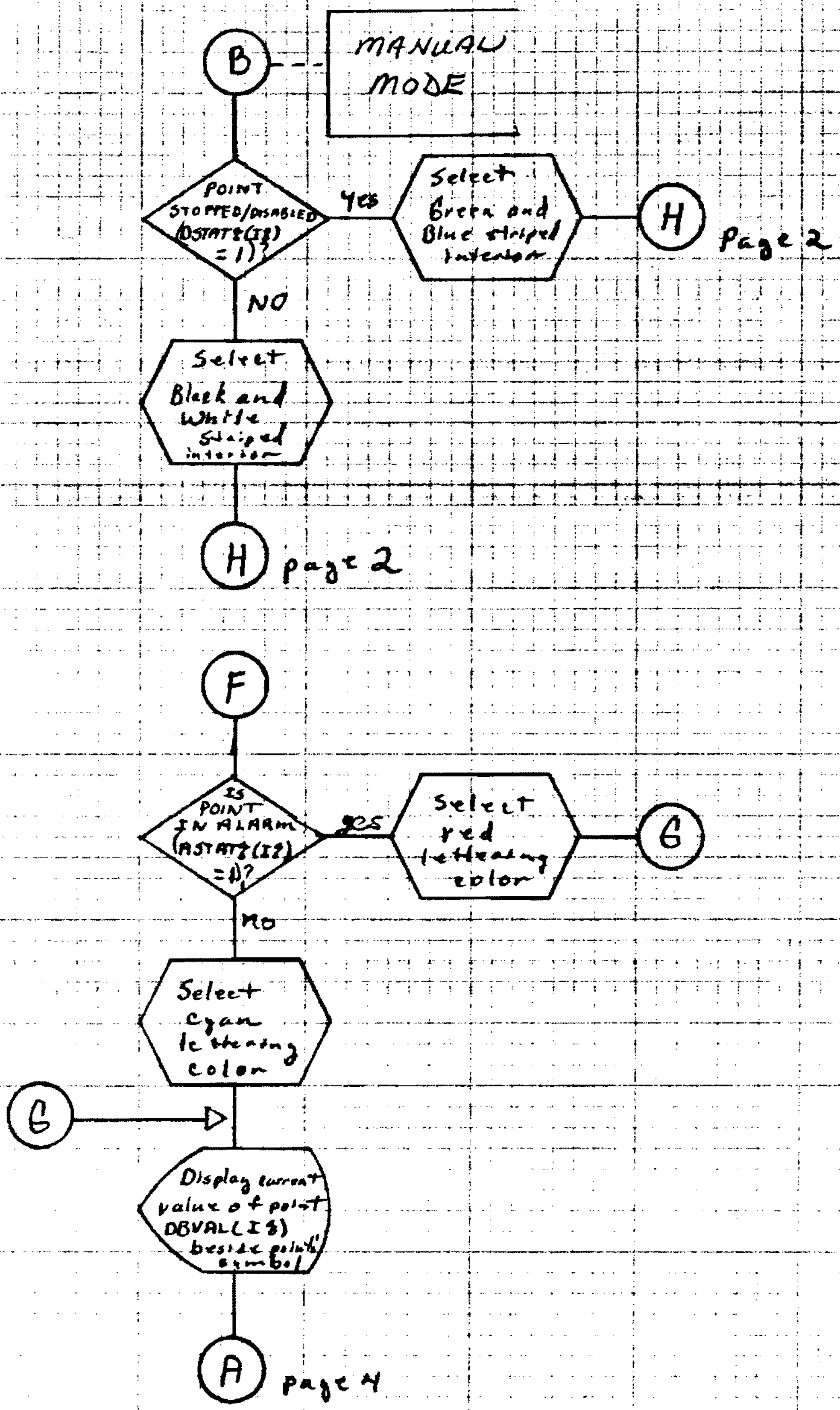
CRT DISPLAY - Display graphics and point values

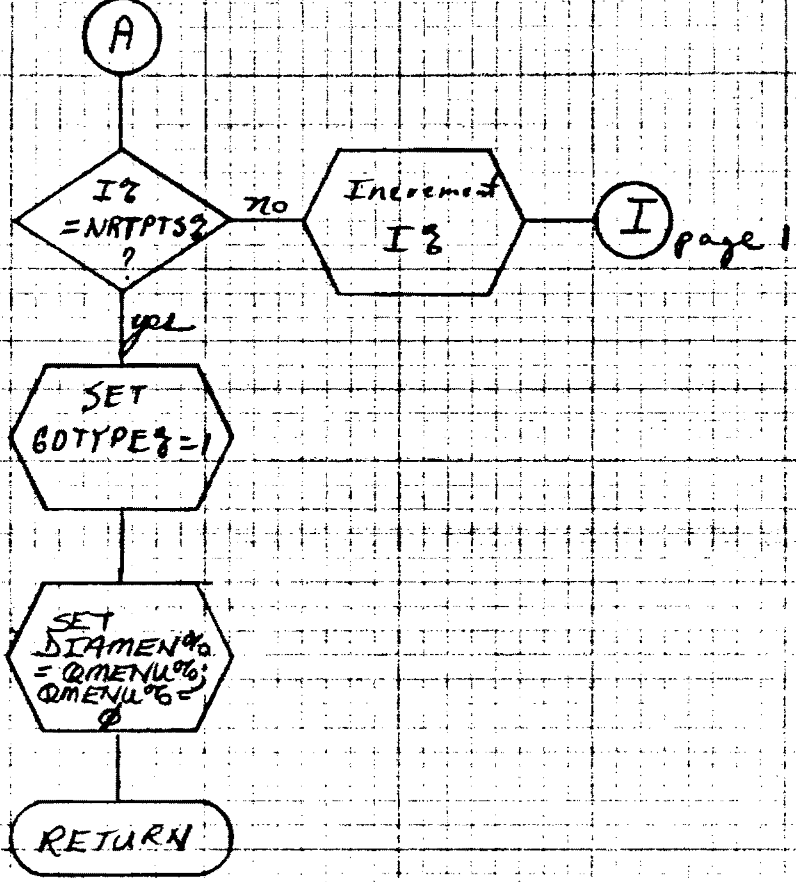
DESIGN NOTES:

(none)


XDD
START
Build filename "DIA" + STR$ (QMENU%) STORE in TMP$
Retrieve File TMP$ from disk Store in buffer memory.
Display buffer memory in graphics area
This buffer file will contain the connecting lines for the symbols, etc.
Set I% = 1
Will loop for number of real-time points in system
I
Point in DE (LPOINT%(I%) = DIAMEN%) ?
NO
A
page 4
yes
DE in Auto mode (DE% = 1)?
NO
B
page 3
yes
C
page 2


C
POINT IN MANUAL (AMPT$(I$) = 0)?
YES
B
page 3
NO
AUTO MODE
POINT STOPPED/DISABLED (DSTAT$(I$) = 1)?
YES
Select Black Interior color
D
NO
Select Green Interior color
D
Is Point the current alarm point (APT$ = I$)?
YES
Select Flashing Symbol
E
NO
E
Select Blue Border color
H
DRAWIT
DRAW SYMBOL IN CORRECT location using colors specified
LOCATION SPECIFIED BY (PX%, PY%)
POINT STOPPED/DISABLED (DSTAT$(I$) = 1)?
YES
A
page 4
NO
F
page 3


B
MANUAL MODE
POINT STOPPED/DISABLED (DSTAT$(I$) = 1)?
YES
Select Green and Blue striped interior
H
Page 2
NO
Select Black and White striped interior
H
page 2
F
IS POINT IN ALARM (ASTAT$(I$) = 1)?
yes
Select red lettering color
G
NO
Select cyan lettering color
G
Display current value of point DBVAL(I$) beside point's symbol
A
page 4


A
I%
=NRTPTS%
?
no
Increment
I%
I
page 1
yes
SET
GDTYPE%=1
SET
DIAMEN%
= @MENU%;
@MENU%=
Ø
RETURN

NAME: XSPL

PURPOSE:

The XSPL subroutine is used to execute the SET POINT/LIMITS function.

OPERATIONAL DESCRIPTION:

The XSPL subroutine determines whether the selected point is analog or digital. If the point is digital, XSPL replaces the current data base alarm value with the one stored in the temporary storage variable ATMP. If the point is analog, XPSL replaces the current data base values for the low limit, high limit, and set point with those stored in the temporary storage variables LTMP, HTMP, and STMP, respectively.

CALLED BY:

CONFIRM

CALLS:

DISMESS

PASSED ARGUMENTS:

ATMP - New digital alarm value
HTMP - New analog high point
LTMP - New analog low limit
PAD% - Analog or digital point discriminator
QPT% - Selected point
STMP - New analog set point

NAME: XSPL

RETURNED ARGUMENTS:

- ALARMVAL - Current digital alarm value
- HILIM - Current analog high limit
- LOWLIM - Current analog low limit
- QPT% - Selected point
- SPT - Current analog set point

FILE INPUT/OUTPUT:

(none)

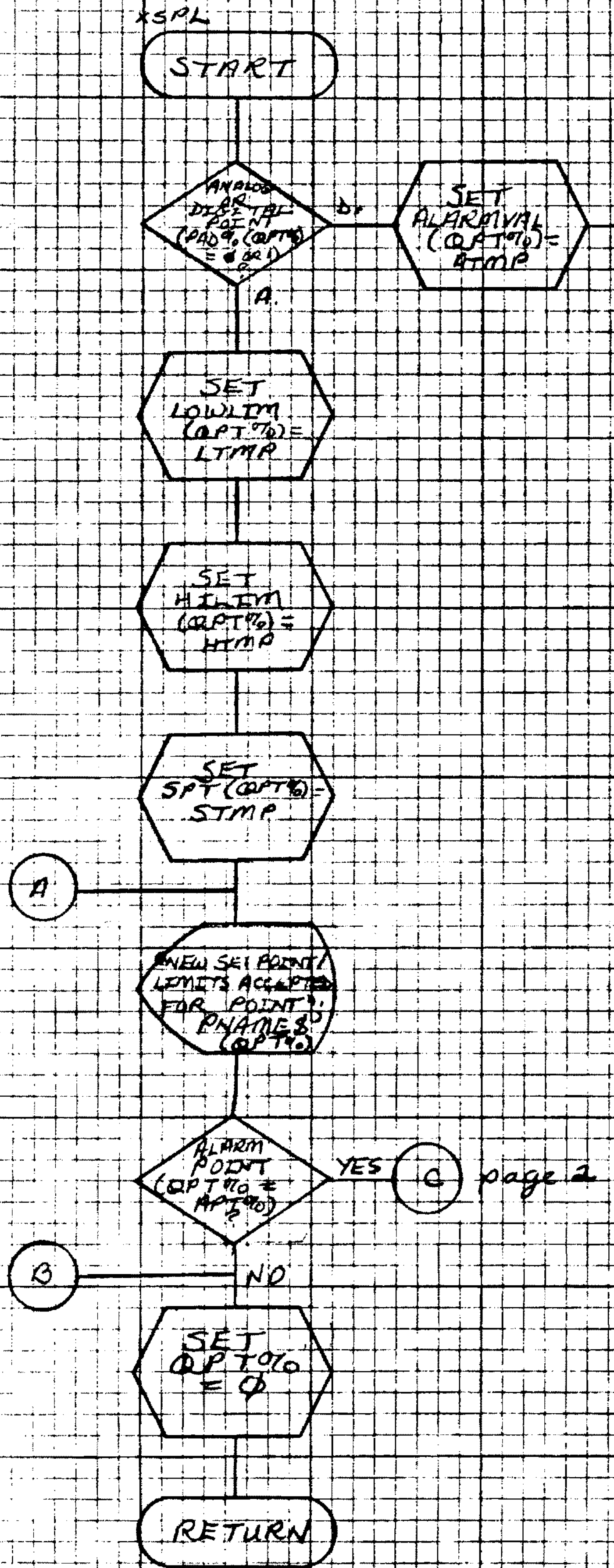
HARDWARE INTERACTION:

(none)

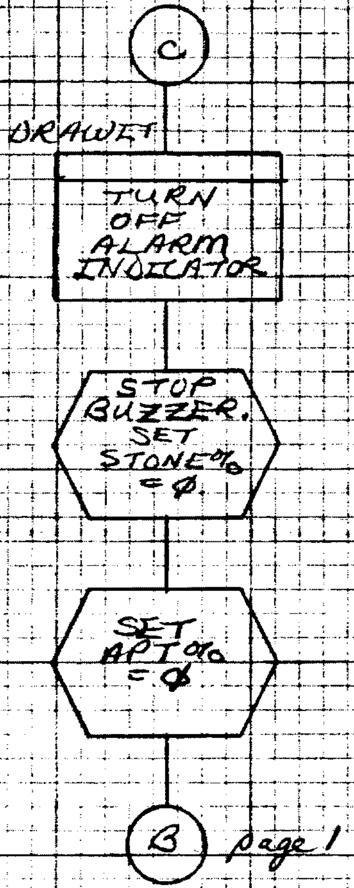
DESIGN NOTES:

The temporary storage variables hold the new values retrieved by the KEYIN subroutine. These values will be stored as new operational values when CONFIRM ACTION is pressed.

XSPL: EXECUTE SET POINT/LIMITS



XSPL: EXECUTE SET POINT/LIMITS



NAME: XAM

PURPOSE:

The XAM subroutine executes the AUTO/MANUAL function.

OPERATIONAL DESCRIPTION:

The XAM subroutine first determines whether the function is to be executed for the entire data environment or just a point. It then updates the data base to reflect the new mode of operation based on the value of QAM%. The screen is also updated to reflect the mode changes.

CALLED BY:

CONFIRM

CALLS:

DRAWIT

PASSED ARGUMENTS:

DIAMEN% - Number of diagram or menu currently displayed

GDTYPE% - Graphics display type

LPOINT% - Number of diagram on which point is located

NDE$ - Textual name of DE

PNAME$ - Textual name of point

QAM% - Selected mode

QPT% - Selected point

NAME: XAM (continued)

RETURNED ARGUMENTS:

AMPT% - Auto/manual mode indicator for point

DE% - Auto/manual mode indicator for DE

QAM% - Selected mode

QPT% - Selected point

FILE INPUT/OUTPUT:

(none)

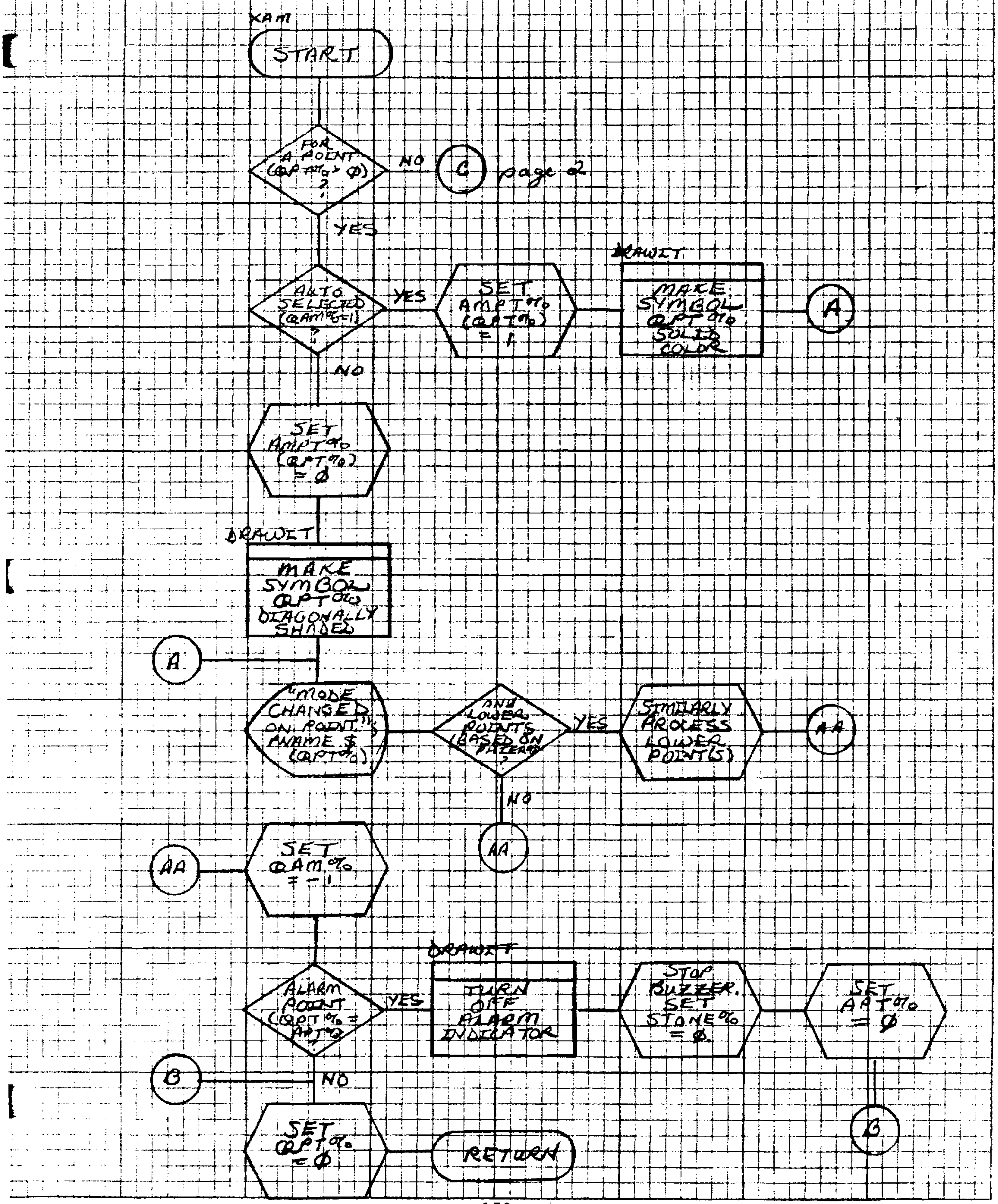
HARDWARE INTERACTION:

CRT DISPLAY - Print messages

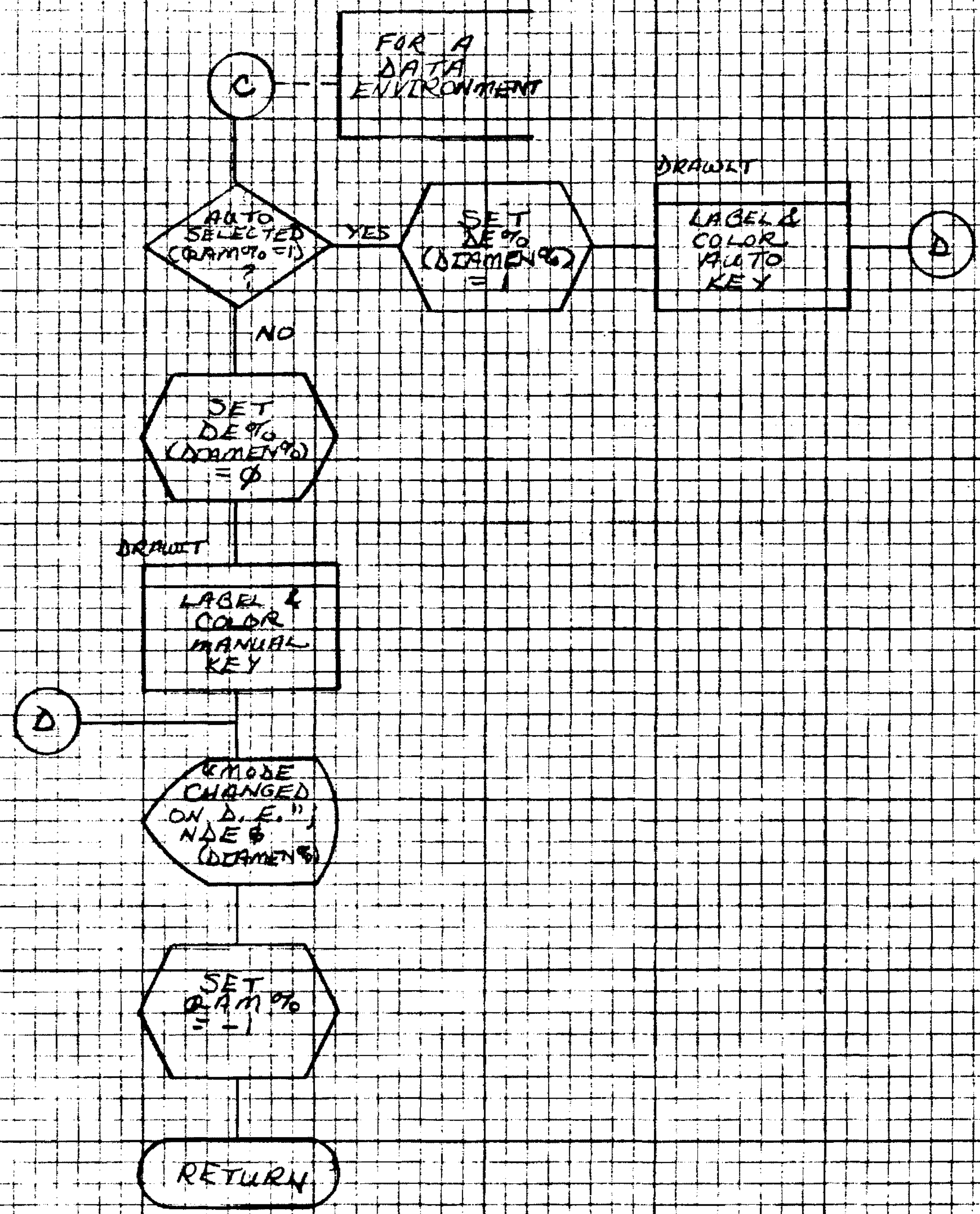
DESIGN NOTES:

(none)

XAM: EXECUTE AUTO/MANUAL



XAM: EXECUTE AUTO/MANUAL



NAME: XPR

PURPOSE:

The XPR subroutine is used to execute the PRINT REPORT function.

OPERATIONAL DESCRIPTION:

A filename is constructed which corresponds to the disk file containing the desired report. This file is then retrieved and sent to the printer.

CALLED BY:

CONFIRM

CALLS:

(none)

PASSED ARGUMENTS:

QMENU% - Selected menu item

RETURNED ARGUMENTS:

QMENU% - Selected menu item

FILE INPUT/OUTPUT:

Files are retrieved from disk and sent to the printer. Filenames are of the form RPT*, where * is the number of the desired report.

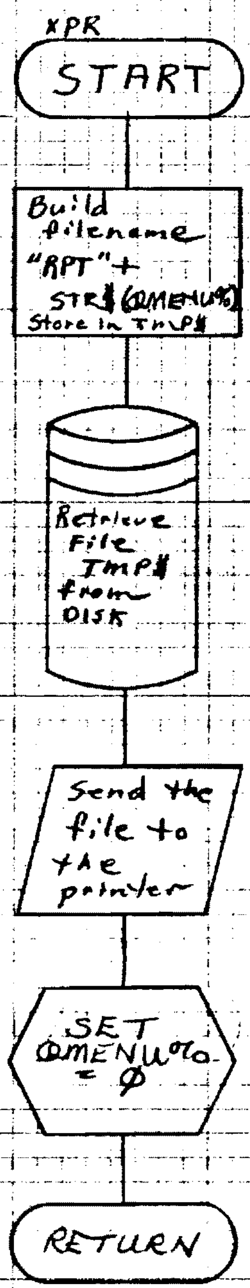
NAME: XPR (continued)

HARDWARE INTERACTION:

PRINTER - Print reports

DESIGN NOTES:

(none)


XPR
START
Build filename "RPT" + STR$(@MENU%) Store in TMP$
Retrieve File TMP$ from DISK
Send the file to the printer
SET @MENU% = Ø
RETURN

NAME: XMS

PURPOSE:

The XMS subroutine is used to execute the MODIFY SCHED function.

OPERATIONAL DESCRIPTION:

The XMS subroutine replaces the current real-time data base schedule for the selected DE with the schedule as modified by the operator.

CALLED BY:

CONFIRM

CALLS:

(none)

PASSED ARGUMENTS:

QSCH% - Number of the data environment schedule to be modified

SCTMP - Temporary storage for new (modified) schedule

RETURNED ARGUMENTS:

S(QSCH%) - Current real-time schedule for DE

QSCH% - Number of DE schedule to be modified

FILE INPUT/OUTPUT:

(none)

NAME: XMS (continued)

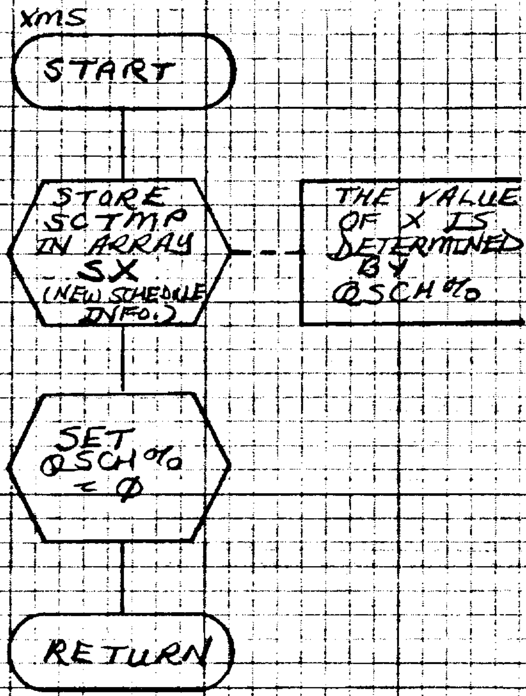
HARDWARE INTERACTION:

(none)

DESIGN NOTES:

(none)

xms: EXECUTE "MODIFY SCHED"



NAME: DISMESS

PURPOSE:

The DISMESS subroutine is used to display color coded messages from the master message file in the message/text area of the CRT display unit.

OPERATIONAL DESCRIPTION:

The DISMESS subroutine first gets the message text from the master message file on disk using MINDEX% as the pointer to the record containing the desired message. It then selects the appropriate color for the message based on the value of MCOLOR%. The message is printed on the display and program control returns to the calling routine.

CALLED BY:

ANNUNC	DIAGRAM	OPER	XSPL
AUTO	FKEYIN	REPORT	
BUZOFF	GONOGO	SCHED	
CANCEL	KEYIN	SETPT	
CONFIRM	MMI EXEC	SYMBIN	

CALLS:

(none)

PASSED ARGUMENTS:

(none)

RETURNED ARGUMENTS:

(none)

NAME: DISMESS (continued)

FILE INPUT/OUTPUT:

Read message text from random access disk file.

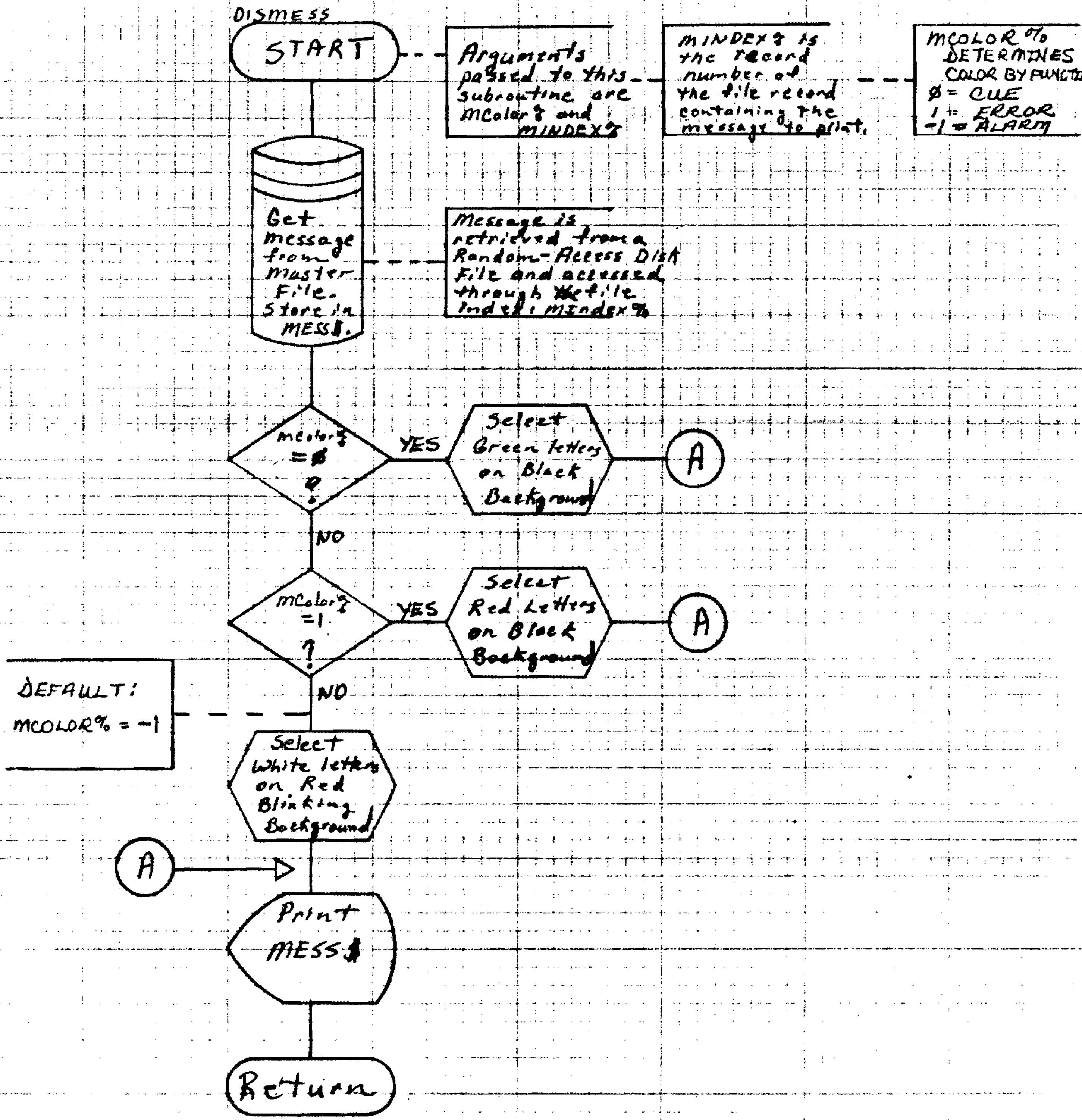
HARDWARE INTERACTION:

CRT DISPLAY UNIT - displays messages

DESIGN NOTES:

(none)

DISMESS: DISPLAY MESSAGES IN THE TEXT AREA FROM THE MASTER MESSAGE FILE



NAME: DMENU

PURPOSE:

The DMENU subroutine is used to place the appropriate menu on the screen.

OPERATIONAL DESCRIPTION:

The DMENU subroutine constructs a string of characters which corresponds to the name of a disk file which contains the desired menu. This file is then retrieved from disk and displayed on the screen. If the menu is for the SET POINT/LIMITS or MODIFY SCHED function, the current set point and limits (or alarm value if a digital point) or the current schedule is also displayed on the screen. Program control then passes to the calling routine.

CALLED BY:

DIAGRAM
REPORT
SCHED
SETPT
SYMBIN

CALLS:

(none)

PASSED ARGUMENTS:

ALARMVAL - Digital point alarm value
DIAMEN% - Number of diagram or menu currently displayed
HILIM - High limit for analog point
LOWLIM - Low limit for analog point
PAD% - Digital vs. analog point discrimination

NAME: DMENU (continued)

PASSED ARGUMENTS: (continued)

QPT% - Seleted point

S(QSCH%) - Schedule for DE

SPT - Set point for analog point

RETURNED ARGUMENTS:

(none)

FILE INPUT/OUTPUT:

Retrieves disk files which contain contents of the memory buffer used to display a given menu on the screen. Filenames are the form MEN**, where ** is a one or two digit number.

MEN1 - Corresponds to the data environment diagram menu

MEN21- Corresponds to the menu skeleton for the SET POINT/LIMITs function for an analog point

MEN20- Corresponds to the menu skeleton for the SET POINT/LIMITs function for an analog point.

MEN3 - Corresponds to the menu of available reports

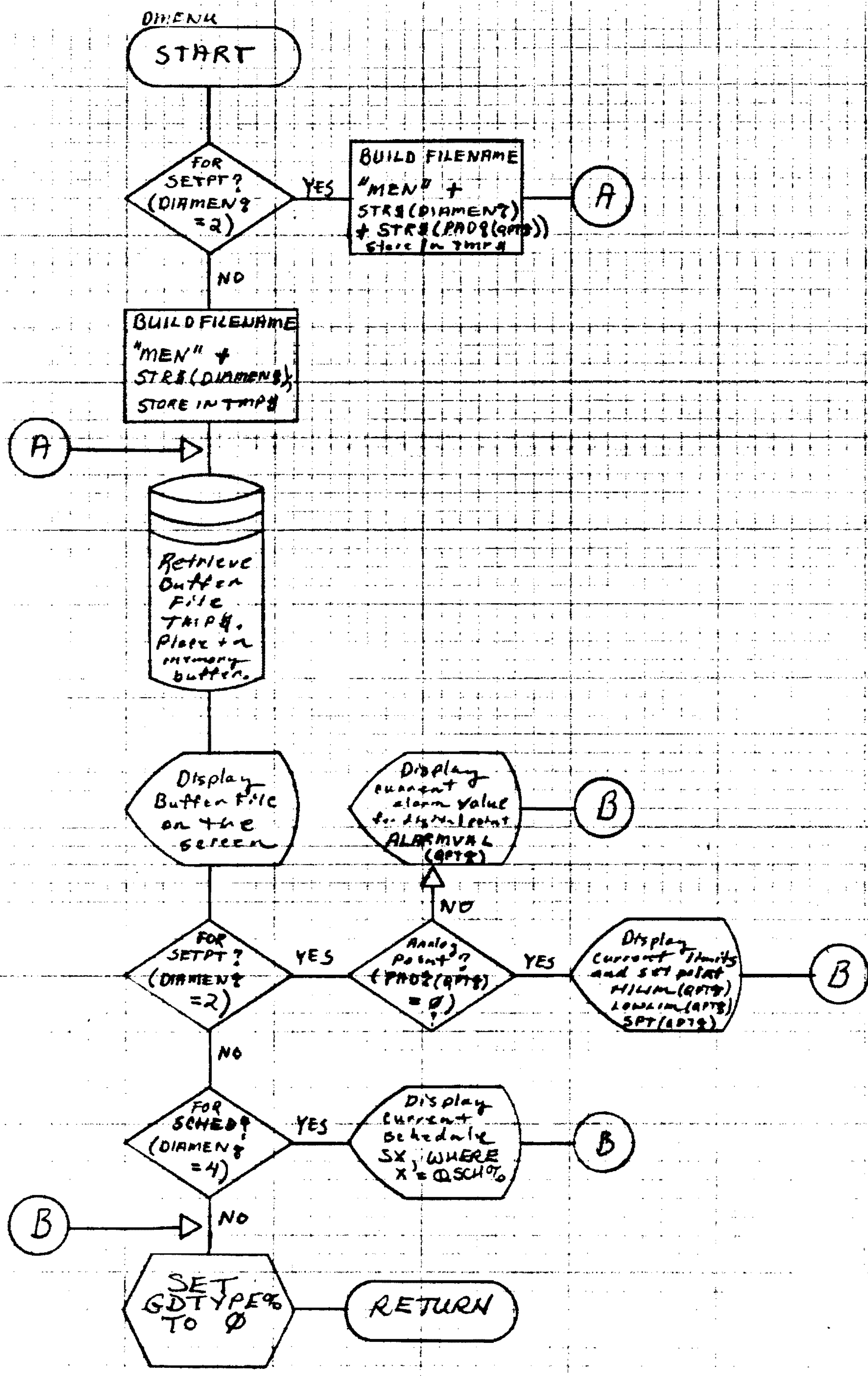
MEN4 - Corresponds to the menu skeleton for the MODIFY SCHED function

HARDWARE INTERACTION:

CRT DISPLAY - Display buffer files and point data

DESIGN NOTES:

(none)


DMENU
START
FOR SETPT? (DIAMEN% =2)
YES
BUILD FILENAME "MEN" + STR$(DIAMEN%) + STR$(PAD%(QPT%)) Store in TMP$
A
NO
BUILD FILENAME "MEN" + STR$(DIAMEN%), STORE IN TMP$
A
Retrieve Buffer File TMP$. Place in memory buffer.
Display Buffer File on the screen
Display current alarm value for digital point ALARMVAL (QPT%)
B
NO
FOR SETPT? (DIAMEN% =2)
YES
Analog Point? (PAD%(QPT%) = Ø)
YES
Display Current limits and set point HILIM(QPT%) LOWLIM(QPT%) SPT(QPT%)
B
NO
FOR SCHED? (DIAMEN% =4)
YES
Display Current Schedule SX, WHERE X = QSCH%
B
B
NO
SET GDTYPE% TO Ø
RETURN

NAME: DRAWIT

PURPOSE:

The DRAWIT utility subroutine is used to draw and redraw HVAC symbols, function keys, menu items, and the alarm indicator. Parameters are used to select interior color, border color, flashing, striping, and rotation.

OPERATIONAL DESCRIPTION:

The DRAWIT subroutine selects the options specified by the parameters and uses the graphics capability of the computer to draw the object selected on the screen.

CALLED BY:

ANNUNC	MDIN	SETPT
AUTO	MLIN	SYMBIN
CANCEL	MRIN	XAM
CONFIRM	MSIN	XDD
DIAGRAM	OPER	XSD
FKEYIN	REPORT	XSE
GONOGO	SCHED	XSPL

CALLS:

(none)

PASSED ARGUMENTS:

D1% - Type of item

D2% - Symbol, menu, or function key number

D3% - Change interior, border, or both

D4% - Color

D5% - Rotation

D6% - Blink or stripe

NAME: DRAWIT (continued)

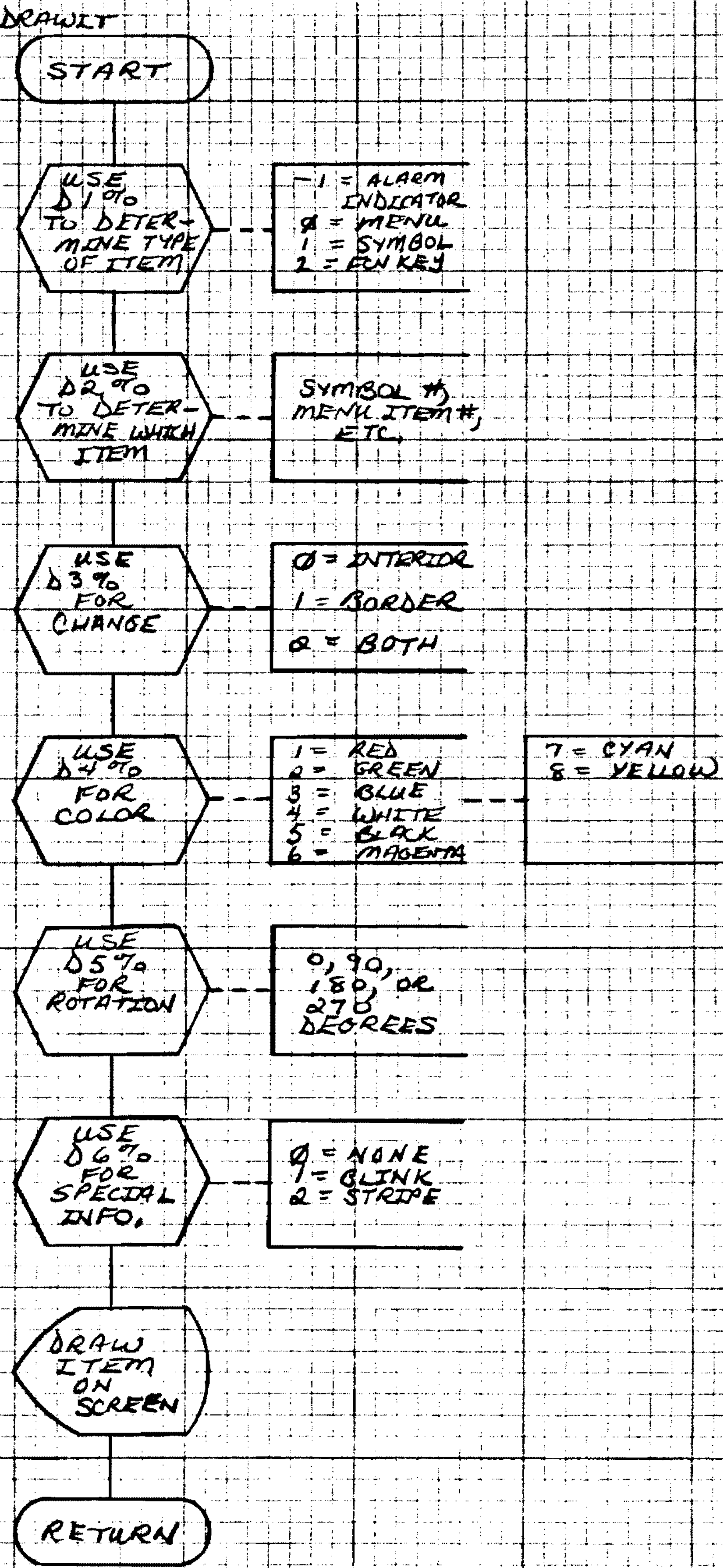
RETURNED ARGUMENTS:
(none)

FILE INPUT/OUTPUT:
(none)

HARDWARE INTERACTION:
CRT DISPLAY - Draws graphics

DESIGN NOTES:
(none)

DRAWIT: DRAW OR CHANGE SCREEN ITEM



APPENDIX A

EQUIPMENT LIST

APPENDIX A

EQUIPMENT LIST

This apppendix contains a listing of the hardware used to implement the MMI demonstration device, along with available specifications for each item.

Color Graphics Computer - Chromatics Model CG3999

GENERAL

Power

105-125 volts, 50/60 Hz, 600 watts.

TEMPERATURE

+10°C to +40°C Operating.

-30°C to +70°C Storage.

HUMIDITY

0-95%, noncondensing.

PACKAGE COLOR

Light Beige (Federal Standard 26521)

and Brown (Federal Standard 20140).

X-RADIATION

Less than 0.5 milliroentgen/hour at a distance of 2 inches from all exterior surfaces.

Color Graphics Computer - Chromatics Model CG3999

PROCESSOR MEMORY

CPU

Z-80

REFRESH MEMORY

Standard Resolution Model - 98,304 bytes dynamic RAM

PROGRAM MEMORY

8,192 bytes of EPROM and 1,024 bytes of RAM for base routines and CRT Operating System.

SOFTWARE

PROCESSORS

CRT Operating System with full, local, and half duplex modes as well as ESCAPE code processing and device assignment capability.

INTERFACES

SERIAL

Serial I/O port - asynchronous; independently programmable from 110 to 31,250 baud (9,600 baud highest standard rate); 1.5 stop bits, programmable to 1 or 2; TTL and RS-232C interface with busy output lines (clear-to-send status line also included in RS-232C interface).

DISPLAY FUNCTIONS

GRAPHIC MODE

DOT, Incremental DOT, X BAR, Incremental X BAR, Y BAR, Incremental Y BAR, Vector, Concatenated Vector.

ALPHANUMERIC MODE

ASCII and special characters available from RAM or EPROM positioned to any dot position on screen.

COORDINATE ENTRY

Relative and absolute modes of decimal digits, binary codes, or cursor position.

CHARACTER FORMAT

96 ASCII upper and lower case 5x7 dot matrix characters, and 96 special 6x10 dot matrix graphics characters programmed in EPROM memory.

CHARACTER MAGNIFICATION

Individually selectable in X or Y to any integer multiplier.

CURSOR

Programmable in color, position, and visibility (on or off).

CHARACTER INTERLINE SPACING

Variable, up to 255 raster lines.

CHARACTER WRITE DIRECTION

Vertical or horizontal.

CHARACTER RESOLUTION

512 x 512 dot resolution - 85 characters/line by 51 lines/page.

CONTROL FUNCTIONS

Cursor Up, Down, Left and Right (character and dot spacing), Erase Page, Erase Line, Home, Tab, Carriage Return, Line Feed, and Backspace.

WINDOWS

4 each, individually programmable in size, position, and all the previously described functions including a separate cursor for each window. Each window is a physical output device and may receive data from any physical input device or logical output device when properly assigned.

VIDEO DISPLAY

Diagonal Measure		PHOSPHOR SIZE		Phosphor Area	
(IN)	(CM)	(IN)	(CM)	(IN^2)	(CM^2)
19	48.3	11.70x15.61	29.7x39.6	182.67	1178

SCREEN SIZE

USABLE DISPLAY AREA

(IN)	(CM)	(IN^2)	(CM^2)	FORMAT
10.25x14.50	26.0x36.8	148.63	958.9	512x512

REFRESH RATE

60 times/second non-interlace, synchronized to 60 Hz line frequency or 50 times/second non-interlaced, synchronized to 50 Hz line frequency.

COLOR LEVELS

8 foreground and 8 background - red, green, blue, magenta, cyan, yellow, white, and black.

CONVERGENCE

9 sector, with each sector individually converged from front drawer accessible controls.

CONTROLS

Brightness, focus, convergence, on/off.

DEFLECTION

Magnetic.

Color Graphics Computer - Chromatics Model CG3999 (concluded)

FOCUS

High voltage electrostatic.

BLINK RATE

1.9 Hz, cursor and dots, foreground and/or background.

CURSOR

4 each, 1 per window, programmable in position, color and visibility (on or off). Expands vertically to match character height when character Y magnification is not equal to 1.

Options added to the basic CG3999:

Option 15 - Expanded card cage
Option 17 - Blink
Option 83 - Keyboard
Option 23 - Memory card w/32k bytes of RAM
Option 31 - Second serial I/O port
Option 41 - Floppy disk controller w/DOS software
Option 43 - Dual standard floppy disk
Option 61 - CPU operating system
Option 62 - Text editor
Option 63 - Z-80 assembler
Option 64 - BASIC language interpreter
Option 76 - Complex boundary fill/pattern
Option 77 - Extended graphics/alphanumerics/zoom
Option 56 - Light Pen
Option 39 - GPIO
Option 38 - 3rd and 4th serial port
Option 65 - Function key processor
Option 54 - Programmable timer

Touch Panel - Carroll 19" IR Touch Input Kit:

CRT Display Size	19"
Matrix	
X Axis	56
Y Axis	48
Resolution	
X Axis	.275
Y Axis	.250
Touch Active Area	
X Axis	15.125
Y Axis	11.750

Scan Rate 40 - 60 CPS depending on size

Data Format Four ASCII characters are generated per HIT

1. Uncover character (identifies Touch Data)
2. X data character
3. Y data character
4. Stop code

Baud Rate Selectable on power box (300, 600, 1200, 2400, 4800, 9600)

Touch Panel - Carroll 19" IR Touch Input Kit (concluded)

Touch Input Features

- All Solid State construction
- Microprocessor Controller
- No obstruction of viewing area
- High reliability
- RS232 interface

Output Printer - NEC Model 7715 Spinwriter:

SPINWRITER TERMINAL SPECIFICATIONS			
Print Speed	55 characters per second (maximum)		
Character Set	128 characters (maximum)		
Print Line	136 columns at 10 characters per inch 163 columns at 12 characters per inch		
Paper Width	16 inches (maximum)		
Paper Thickness	.027 inch (maximum)		
Carriage Return Time	400 milliseconds (maximum)		
Column Spacing	1/120 positions per inch		
Line Spacing	1/48 positions per inch		
Line Feed	6 or 8 lines per inch		
Forms Length	1-99 lines		
Forms Slew Rate	4.16 inches per second plus 53 milliseconds Settling Time		
Interface Type	EIA RS-232C/CCITT V.24 and 20/60 milliamp current loop		
Communication Code	ASCII		
Communication Modes	Half or full duplex		
Transmission Speed	110, 300 or 1200 baud (standard) 150, 200 or 600 baud (optional)		
Parity	Odd, Even, Marking Bit		
Receive Buffer	256 characters		
Keyboard Buffer	16 characters		
Standard Protocol	ETX/ACK, DC1/DC3 (XON-XOFF) or Reverse Channel		
Modem Compatibility	Bell 103A, 103E, 103F, 103G, 103H, 113A (or equivalent), or Acoustic Coupler		
Overall Dimensions		5515-RO	5525-KSR
	Width	24.8 inches	24.8 inches
	Depth	16.34 inches	21.1 inches
	Height	8.68 inches	8.68 inches
Weight	45.5 pounds (20.7 Kg) 51 pounds (23.2 Kg)		
Power Requirement	115 vac ±15%, 3.5 amps, 50/60 Hz or 230 vac ±15%, 2 amps, 50/60 Hz		
Power Consumption	80 watts (standby) 180 watts (operating)		
Temperature	Operating 40° F (5° C) to 100° F (38° C) Storage −4° F (−20° C) to 158° F (70° C)		
Relative Humidity	Operating 10% to 85% (no condensation) Storage 10% to 95% (no condensation)		
Altitude	Operating sea level to 10,000 feet Storage sea level to 25,000 feet		
Acoustic Noise	60 dBA (with covers)		

Options added to the basic 7715:

Cut-sheet feeder

Vertical forms tractor

APPENDIX B

CUES, ALARMS, and ERROR MESSAGES

APPENDIX B

CUES, ALARMS, and ERROR MESSAGES

All cue, error, and alarm, messages are printed in color on the screen using the following color code scheme:

- Cues - printed in green
- Errors - printed in red
- Alarms - printed in white on a red blinking background.

This Appendix follows the same color coding convention in that the three types (colors) of messages are broken into separate lists. Also, the messages in each list have been alphabetized.

CUES - GREEN LIST

Cue Message	Explanation
- Auto Mode Selected For DE	If the currently displayed DE is in Manual Mode and the AUTO/MANUAL key is touched, this cue appears to notify the operator that the automatic mode has been requested for this DE. The computer will not execute the mode change command until the CONFIRM ACTION key is touched.
- Auto Mode Selected For Point (x)	If the point selected is in Manual Mode and the AUTO/MANUAL key is touched, this cue appears to notify the operator that the automatic mode has been requested for this point. The computer will not execute the mode change command until the CONFIRM ACTION key is touched.
- Command Action Cancelled	This cue appears when the operator touches the CANCEL ACTION key while entering a command sequence. The command sequence is aborted and the screen returns to its previous state.
- Command Action Completed	After the CONFIRM ACTION key is touched, the system will attempt to execute the command. If it is successful, this message will be given.
- Device (x) Has Been Started/Enabled	This cue signals the operator that the EMCS computer has started/enabled device x, where x is a device which the operator manually commanded the system to start.
- Device (x) Has Been Stopped/ Disabled	This cue signals the operator that the EMCS computer has stopped/disabled device x, where x is a device which the operator manually commanded the system to stop.

CUES - GREEN LIST (continued)

Cue Message	Explanation
- Enter Password	This cue appears during the change operator command sequence upon successful acceptance of the operator's name. At this point, the operator's password should be typed in using the keyboard.
- Manual Mode Selected For DE	If the currently displayed DE is in automatic mode and the AUTO/MANUAL key is touched, this cue appears to notify the operator that the manual mode of operation has been requested for this DE. The computer will not execute the mode change command until the CONFIRM ACTION key is touched.
- Manual Mode Selected For Point (x)	If the point selected is in automatic mode and the AUTO/MANUAL key is touched, this cue appears to notify the operator that the manual mode has been requested for this point. The computer will not execute the mode change command until the CONFIRM ACTION key is touched.
- Mode Changed On Point (x)	The operator has used the AUTO/MANUAL function to change the mode of operation for point (x). The system is acknowledging the fact that it has processed the mode change request.
- Mode Changed On DE (x)	The operator has used the AUTO/MANUAL function to change the mode of operation for DE (x). The system is acknowledging the fact that it has processed the mode change request.

CUES - GREEN LIST (continued)

Cue Message	Explanation
- New Device Selected	This cue appears if the operator touches a second device symbol when one has already been selected. The old choice is deselected and the new choice is highlighted.
- New Function Selected	This cue appears if the operator touches one of the first eight keys when one has already been selected. The old choice is deselected and the new choice is backlighted.
- New Menu Item Selected	This cue appears if the operator touches a second menu item when one has already been selected. The old choice is deselected and the new choice is highlighted.
- New Set Point/Limits Accepted For Point (x)	The operator has used the SET POINT/LIMITS function to change the set point and/or limits for device (x). The system is acknowledging the fact that it has processed the new values and added them to the data base.
- Or Touch Device, To Set Mode For One Point	The operator has touched the MODIFY SCHED key and a point has not been selected. The computer will assume that the mode change will be made for the DE unless a device is selected.
- Password Accepted. New Operator Is (x).	If the password matches the one approved for the new operator during the CHANGE OPER command, this cue appears in the text area and the Date/Time/Operator display is updated. X is the new operator's name.

Cue Message	Explanation
- Please Type In The New Value (Or Touch CONFIRM ACTION)	This cue appears during the SET POINT/LIMITS and MODIFY SCHED command sequence. It instructs the operator to type in a new value for the parameter that is to be modified or, if all values appear as desired, to touch the CONFIRM ACTION key to execute the changes.
- Ready To Accept Command	This cue appears whenever the MMI is idle and waiting for the operator to enter a command sequence. The cue is not imperative, but rather it is simply used to indicate that the MMI computer is operational and ready to be used.
- Ready To Change Operation - Enter Operator Name, HELP, or STOP From Keyboard	These two cues are displayed when the operator touches the CHANGE OPER key. At this point, the operator should type in the desired command followed by the Return key using the keyboard.
- Ready To Modify Schedule	When the MODIFY SCHED command is entered, this cue appears in the text area to inform the operator that the computer is ready to make the desired schedule changes for the current DE.
- Ready To Modify Set Point And/Or Limits	When the SET POINT/LIMITS key is touched, this cue appears in the text area to inform the operator that the computer is ready to make the desired analog limit changes to the device selected.

CUES - GREEN LIST (continued)

Cue Message	Explanation
- System Shutdown	The operator has touched the CHANGE OPER function key and typed in the command STOP from the keyboard. This message indicates that the system has been turned off (stopped, shutdown) by the operator.
- Touch Appropriate Device Symbol	When a function key requiring a device selection is touched prior to the device symbol, this cue appears instructing the operator to select the appropriate device.
- Touch CONFIRM ACTION To Execute	This cue appears in the text area when all of the parts of a command sequence have been entered. If CONFIRM ACTION is touched, the computer will execute the command.
- Touch CONFIRM ACTION When All Values Appear As Desired	This cue appears in the text area when either the SET POINT/LIMITS or MODIFY SCHED function is selected. It is used to remind the operator that the computer will not execute any changes to the schedule or to control values until the CONFIRM ACTION key is touched.
- Touch Desired Function	When a device symbol is touched prior to a function, this cue appears instructing the operator to touch the desired function.
- Touch Square Beside Building/Floor Plan Desired	When the DISPLAY DIAGRAM key is touched, a menu of available Data Environments appears in the graphics area. This cue instructs the operator to select which DE he wishes to have displayed.

CUES - GREEN LIST (concluded)

Cue Message	Explanation
- Touch Square Beside Desired Report	When the PRINT REPORT key is touched, a menu of available reports appears on the screen. This cue appears in the text area instructing the operator to select the desired report to be printed.
- Touch Square Beside Desired Value To Be Changed	The operator has entered either the MODIFY SCHED or SET POINT/LIMITS command. To modify a value, it must be selected. This is done by touching the square beside the desired value.
- Type RETURN To Continue Or S RETURN To Stop	The operator is examining the HELP routine. This message is a prompt signaling that the system is waiting for a response from the operator before continuing with the next screen of HELP information. If the operator types RETURN, the next screen will be displayed. If the operator types S RETURN, then the HELP sequence will be exited.

ERRORS - RED LIST

Error Message	Explanation	Corrective Action
- Command Inappropriate For Device, No Action Taken	A command sequence cannot be executed because the command is not appropriate for the device. For example, a fan does not have analog values. An attempt to set analog values for the fan will yield this error message.	Enter a new command sequence.
- Device Already Started/Enabled, No Action Taken	The operator has attempted to manually stop or disable a device which is already stopped/ disabled.	Select a new device or function.
- Device Already Started/Enabled, No Action Taken	The operator has attempted to manually start or enable a device which is already started/ enabled.	Select a new device function.
- Device Already Stopped/Disabled, No Action Taken	The operator has attempted to manually stop or disable a device which is already stopped/ disabled.	Select a new device
- Grievous System Error	This message indicates an internal fault of the system. It should never appear under normal circumstances.	Continue processing from point of error.
- High Limit Can Not Be Lower Than Low Limit, Please Reenter	The operator has attempted to set the high limit for an analog device lower than the current low limit using the SET POINT/LIMITS command.	Enter a value for the high limit which is greater than the current low limit, or reset the low limit to a value which is less than the new high limit.

Error Message	Explanation	Corrective Action
- Higher Level Is In Manual Mode, No Action Taken	The operator has attempted to place a point in automatic mode when its supporting device is in manual mode.	Change mode of higher device first, or select a new point.
- Incomplete Command Sequence, Please Continue	The operator has touched the CONFIRM ACTION key before the entire command sequence was entered. For example, the operator requested the STOP function when no device was selected.	The operator may now continue entering the command.
- Low Limit Cannot Be Higher Than High Limit, Please Reenter	The operator has attempted to set the low limit for an analog device higher than the current high limit using the SET POINT/ LIMITS command.	Enter a value for the low limit which is less than the current high limit or reset the high limit to a value which is greater than the new low limit.
- Name Not Recognized, Command Action Cancelled.	The name typed in for the new operator is unknown to the system.	Enter a new command sequence. If the CHANGE OPER function is to be selected again, check to make sure the operator name is typed correctly before typing the carriage return key. If the error occurs agaian, see the level 2 operator to make certain that the operator name has been placed on the list of approved system operators.
- Password Not Recognized, Access Denied	The password entered does not match the one approved for the operator; therefore, access to the system is denied.	Enter a new command sequence. If the CHANGE OPER function is to be selected again, make certain that the password is entered correctly. If the error occurs again, see the level 2 operator to obtain the correct password.

ERRORS - RED LIST (concluded)

Error Message	Explanation	Corrective Action
- Set Point Not Within Range Limits, Please Reenter	The operator has attempted to change the set point to a value outside of the current range limits for a particular analog device.	Enter a value for the set point that is within the current range. Alternately, change the range.
- Typographical Error, Please Reenter The Data Item.	The operator has made a typographical error while entering data from the keyboard. Typical errors include typing a letter within a number or comma within a number.	Correct the error by retyping the correct item.
- You Cannot Set Points/Limits For That Device--Try Another	The operator has attempted to set analog values for a device which is not analog.	Select an analog device, new function, or CANCEL ACTION.
- You Must Use DISPLAY/DIAGRAM To Get A Diagram Before You Can Use This Command	The operator has selected either the START/ENABLE, STOP/DISABLE, SET POINT/LIMITS, AUTO/MANUAL, or MODIFY SCHED function when no data environment diagram is displayed.	Touch the DISPLAY DIAGRAM key and then the desired menu item to display the DE diagram. Now reenter the desired command.

ALARMS - WHITE ON RED FLASH LIST

Alarm Message	Explanation
- Alarm Acknowledged--Audible Tone Temporarily Silenced	This cue appears when the operator touches the alarm indicator to temporarily silence the audible alarm tone. It signifies that the system has received operator acknowledgment of the alarm.
- Alarm Condition Corrected At (x)	This message signals the operator that the alarm condition no longer exists for point x, where x is a point in the system which was previously in alarm.
- Alarm Condition Detected	When an alarm condition is passed by EMCS, the MMI alarm indicator will be activated, an audible tone will sound, and this message will appear in the text area.
- To Temporarily Silence Tone And Acknowledge Alarm, Touch The Alarm Indicator	This cue appears after the operator has been notified that the alarm condition exists. If the operator so desires, he can touch the alarm indicator to temporarily silence the audible tone.
- Touch DISPLAY DIAGRAM For More Alarm Information	This Cue appears after the operator has been notified that an alarm condition exists. If the operator touches the DISPLAY DIAGRAM key it will cause the menu of data environments to be displayed. This menu will show which area has the alarm, and the operator may then command the computer to display the alarm area for subsequent corrective action.

MAN-MACHINE INTERFACE DEVICE

OPERATOR'S MANUAL

Man-Machine Interface Device

Operator's Manual

by

R.C. Coleburn

B. S. Rice

J.S. Cathcart

B. B. Wise

of the

Command and Control Branch
Electronics and Computer Systems Laboratory
Engineering Experiment Station
Georgia Institute of Technology
Atlanta, Georgia 30332

GIT/EES Project A-3008

May 1982

Prepared for

NAVAL CIVIL ENGINEERING LABORATORY
Port Hueneme, California 93043

TABLE OF CONTENTS

1.0 OPERATION

1.1 Cabling Hookups

There are two special purpose devices used with the standard Chromatics CG 3999 Series Computer. These two devices require special cables to tie them to the computer. All other cabling is the standard set up for a CG 3999 computer. The first device is the NEC Spinwriter Model 7715. A null modem must be connected between the serial interface cable from the Spinwriter and Serial Port # 0 on the backplane of the computer. The connectors required to construct the null modem are a male and a female DB-25 D-shell connector. The female connector plugs into the Serial cable from the printer. The male connector plugs into Serial Port # 0 on the backplane of the computer. The required connections between the male and female connectors are as follows:

DB-25 Female (Spinwriter pin #)	must be connected to	DB-25 male (CG pin #)
1		1
2		3
3		2
4-5 (jumpered together)		
7		7
19		5
6-8-20 (jumpered together)		

Additionally, the NEC Spinwriter should be configured to accept 1200 baud communication rate, run Reverse Channel active HIGH, operate in Full Duplex mode, have Parity Clock Disable, 10 characters per inch, and 6 lines per inch. Further, the Form Length should be set on 66 and the printer operated in the Remote mode. All other switches on the front panel should be in the off (down/0) position. Refer to Figure A-3 in the Operator's Guide for the NEC Spinwriter for more details.

The second device is the Carroll Touch Panel. This device must be connected into Serial Port # 1 on the backplane of the CG 3999 Computer. A four wire cable of approximately four to six feet should be used. The connectors required are two DB-25 D-Shell male connectors. The following pin connections should be made between the Carroll Touch Panel logic box and the CG Serial Port # 1:

DB-25 Male (Carroll Logic Box Pin #)	must be connected to	DB-25 Male (CG pin #)
1		1
2		3
3		2
7		7

Additionally, the Carroll Logic box should be set up to transmit its data at 150 baud with a Start Code of 124, and a Stop Code of 123. Also, the Touch Panel should be operated in the Point Mode. Refer to the Operator's Manual which is provided with the Touch Panel kit for further details.

1.2 Touch Panel Calibration

A separate program has been provided with the EMCS MMI software to calibrate the Carroll Touch Panel. This program performs a great deal of intense numeric calculations, the details of which are not provided here. If more detail is required see Section 3.3 and Appendix C.3 of the Design Manual.

Prior to initial use of the system, and at any time after mechanical adjustment of the touch panel frame, the following procedure should be used to calibrate the touch panel.

1. Turn on power to the computer and disk drive unit 1.
2. Insert a copy of the master disk into drive 1.
3. Press the RESET key.
4. Press the BOOT key.
5. Press the DOS key.
6. Type in: SUBMIT INTSETUP
7. Press the RETURN key.
8. The touch panel recalibration program will now begin. Follow all instructions as they are presented on the screen.
9. When the message "CALIBRATION COMPLETE" appears on the screen, the recalibration is finished.

1.3 Software Boot-Up of EMCS MMI Program

1. Turn on power to the computer, disk drive 1 and the printer.
2. Insert a disk containing the MMI software into drive 1.
3. Depress RESET key for approximately 1/2 second.
4. Depress BOOT key.
5. Depress DISK OS key.
6. Type in: SUBMIT STARTMMI
7. Press the RETURN key.
8. Follow directions on CRT screen.

Alternate boot-up procedure from secondary keyboard (NOTE: all letters must be CAPITAL letters).

1. Turn on power to the computer, disk drive 1 and the printer.
2. Insert a disk containing the MMI software into drive 1.
3. Depress ESC key, SHIFT key, and G simultaneously (BOOT).
4. Depress ESC key, SHIFT key, and D simultaneously (DOS).
5. Type in: SUBMIT STARTMMI.
6. Press the RETURN key.
7. Follow directions on CRT screen.

1.4 Correcting Typing Mistakes

From time to time the operator will be requested to type in a number or name. In so doing, it is not uncommon for a typing mistake to be made. The operator can correct such mistakes before the RETURN key is depressed by backspacing over them. Backspacing is accomplished by pressing the Backarrow key (--) or by holding down the control key (CNTRL) while pressing the H key. Both methods perform the exact same function. When a character on the screen is backspaced over, it will be erased and a new character may be typed in its place or the backspace key may be used to erase another character. When the mistake has been corrected the RETURN key should be pressed to tell the computer to accept the new value or name.

2.0 FUNCTION KEY DESCRIPTIONS

Function key descriptions follow the general pattern of operation of the EMCS/MMI outlined in Section Two of the Design Manual. The description for the START/ENABLE key will include more detail than that of the remaining nine keys so as to allow the operator to become more familiar with the pattern.

2.1 START/ENABLE Key

The START/ENABLE key is used to manually start mechanical devices and enable monitoring and control devices. This key can be operated only when a data environment is in manual mode. Touching the START/ENABLE key when the data environment is in automatic mode will cause the system to display the error message, "Data environment must be in manual mode before this function can be selected." The word "device" is used to refer to any point in the system which can be operated independently, whether it be a FID, MUX, pump, fan, or data point (analog or digital input or output). The START/ENABLE key is colored green, which connotes "go" or "start," to aid the operator in rapidly associating its color with its function and in scanning for the key.

The START/ENABLE key is used in the following manner. When a data environment (DE) operating in manual mode is displayed on the screen, the operator can start or enable any selectable device (indicated by a cyan border) on that DE by touching the START/ENABLE key and the appropriate device symbol on the graphic display (order is independent). When the key is touched, it changes color to yellow (hereafter referred to as backlighting the key), indicating that it has been selected. Likewise, when a device is selected, its border changes color to yellow (hereafter referred to as highlighting), indicating that it has been selected. If the key is touched first, the cue "-Touch Appropriate Device Symbol" appears in text area. If the device symbol is touched first, then the cue "-Touch Desired Function" appears in the text area. Once both a function and a device have been selected, the cue "-Touch CONFIRM ACTION To Execute" appears in the text area. At this point the operator can do one of three things:

1. Touch the CONFIRM ACTION key, at which time it becomes backlighted; the command to start/enable the device is sent to the EMCS; the START/ENABLE key backlight is extinguished; the border of the device symbol reverts back to its original color, and its interior changes to an appropriate color to indicate that it has been started/enabled; the message "-Device (x) Has Been (Started/Enabled)" appears in the text area; and finally, after a slight pause (representing the time needed for the requested action to occur), the CONFIRM ACTION key backlight is extinguished.

2. Cancel the entire operation by touching the CANCEL ACTION key, at which time it becomes backlighted; the START/ENABLE key's backlight is simulaneously extinguished; the device symbol border color returns to its original color; and finally the message "-Command Action Cancelled" appears in the text area.

3. Touch a different device symbol and/or function key, at which time the appropriate symbol border color changes are made and/or the function key backlights are changed, and appropriate cues are given.

2.2 STOP/DISABLE Key

The STOP/DISABLE key is used to manually stop selectable mechanical devices and disable monitoring and control devices when a data environment is in the manual mode. It operates in the same manner as the START/ENABLE key. This key is colored red because red connotes "stop", and this further aids the operator in distinguishing its function.

2.3 DISPLAY DIAGRAM Key

The DISPLAY DIAGRAM key is used to display HVAC diagrams of specific data environments (DE's) in the graphics display area. It is colored blue, which connotes "calmness," since its use does not directly alter the operation of the system. When the key is touched, it becomes backlighted and the building or floor selection menu is presented in the graphics area. The cue "-Touch Square Beside Building/Floor Plan Desired" appears in the text area. The menu is presented in cyan and the squares are color-filled targets. When touched, the square and its text description change color to yellow to indicate which data environment has been selected, and the cue "-Touch CONFIRM ACTION To Execute" appears in the text area. If the operator touches the CONFIRM ACTION key, it becomes backlighted, the DISPLAY DIAGRAM key backlight is extinguished, the selected diagram appears in the graphics area, and finally the CONFIRM ACTION key backlight is extinguished. Alternately, the operator can change the menu or function selections, or touch the CANCEL ACTION key to cancel the command.

2.4 SET POINT/LIMITS Key

The SET POINT/LIMITS key is used to adjust set points and analog limits for selectable analog HVAC devices (indicated by a cyan border). This key is magenta in color. A DE diagram must be present on the screen and in manual mode when this key is touched. When this key is touched it becomes backlighted, and the cue "-Touch Appropriate Device Symbol" appears in the text area. If the symbol was touched prior to touching the key, the cue "-Touch Desired Function" would have appeared in the text area. If the function is not appropriate for the device (i.e., the device is not an analog point), the message "-You Cannot Set Points/Limits For That Device -- Please Try Another" appears in the text area. When both function and device have been selected, the cue "-Touch CONFIRM ACTION to execute" appears in the text area. At this point the operator can change the function selection, or use the CANCEL ACTION key. If the operator touches CONFIRM ACTION, a menu of choices is presented in cyan: one for the set point, one for the low limit, and another for the high limit. Each choice will have a touch-sensitive square beside it. To modify any or all points/limits, the operator merely touches the desired selections, one at a time, and the cue "-Please Type In The New Value" appears in the text area. The choice selected will always change color to yellow. The operator types in the new value and then presses the RETURN key. At this point the operator may modify another value simply by touching its menu square and repeating the steps listed above. When all of the points/limits appear as desired, the operator merely touches CONFIRM ACTION, at which time the graphics area returns to the previous diagram, the key backlights go out, and the message "-Command Action Completed" appears in the text area. Note that the operator could still have decided not to make any changes simply by using the CANCEL ACTION key.

2.5 AUTO/MANUAL Key

The AUTO/MANUAL function key is a special toggle key which serves two purposes. It functions as a status indicator as to which mode the particular data environment (DE) is operating under, as well as a means of selecting the operating mode for the DE. The AUTO/MANUAL function applies only to an entire data environment, rather than to an individual device. Therefore, any device which is selected during operation of this function will be ignored. This key is colored white to make it easily distinguishable from the other function keys (no other key is white) and placed in the middle of the screen in order that it can be easily seen. The key has three different representations on the screen:

1. When no DE is displayed, the key will have the words "AUTO/MANUAL" printed on it in black letters to indicate its function.
2. When a DE is displayed which is running in automatic mode, the key will have the word "AUTO" printed on it in blue letters indicating that the DE's mode of operation is automatic. Blue lettering is used since blue connotes "calmness," and automatic mode is "calm" since it requires no intervention from the operator under normal circumstances.
3. When a DE is displayed which is running in manual mode, the key will have the word "MANUAL" printed on it in red letters indicating that the DE's mode of operation is manual. Red lettering is used because red connotes an abnormal condition, and when in manual mode the operator must take direct action to control the DE.

When a DE is displayed and the key is touched, the key will backlight, and the cues "-Manual Mode Selected For DE" (or "-Auto Mode Selected For DE") and "-Touch CONFIRM ACTION To Execute" will appear in the text area. If CONFIRM ACTION is touched, the message "-Command Action Completed" will appear in the text area, the function key will toggle to indicate the new mode of operation, the backlight will extinguish, and the screen will reflect the current status of the DE. Otherwise, the operator can select a different function or use CANCEL ACTION.

2.6 PRINT REPORT Key

The PRINT REPORT key is used to allow the operator to initiate the display or printing of special reports. It is colored blue because blue connotes "calmness," and use of this key does not disturb the system's operation. Touching this key causes it to be backlighted and a menu of available reports, colored in cyan, to appear on the screen. The message "-Touch Square Beside Desired Report" appears in the text area. Touching a square elicits the cue "-Touch CONFIRM ACTION To Confirm Report Selection," and the choice is highlighted in yellow. If CONFIRM ACTION is touched, then the report menu screen will be erased and a menu of available output devices, colored in cyan, will appear on the screen. The message "-Touch Square Beside Desired Report Output Device" appears in the text area. Touching a square elicits the cue "-Touch CONFIRM ACTION to Execute," and the choice is highlighted in yellow. If CONFIRM ACTION is again touched, the message "-Command Action Completed" is written in the text area, after which the key backlights go out. The requested report is then sent to the printer or the CRT screen in accordance with which output device was chosen. If the report was displayed on the screen, the message "-Press RETURN To Continue..." will appear in the text area. When the RETURN key is depressed, the report will be erased and the screen will return to its previous status. If the report was printed on the printer, the screen will automatically return to its previous status. The operator may select a different report, output device, or function (before CONFIRM ACTION is touched) at which time the screen is updated to reflect the change, or use CANCEL ACTION to inhibit any report.

2.7 MODIFY SCHED Key

The MODIFY SCHED key is used to modify the automatic schedule of operation for a particular data environment (DE). Like the SET POINTS/LIMITS key, it is colored magenta to indicate that these functions are related in that they both modify stored operating parameters. This key may be used to modify the schedule for any data environment. When the MODIFY SCHED key is touched it becomes backlighted, the message "-Ready to Modify Schedule" appears in the text area, and a menu of schedule selections appears on the screen. Next, the message "-Touch Square Beside Desired Schedule To Be Modified" appears in the text area. When a schedule is selected, the message "-Touch Confirm Action to Confirm Schedule Selection" appears in the text area. At this time the operator may either select a new menu item (i.e., the schedule for another DE), touch CONFIRM ACTION, or touch CANCEL ACTION.

When CONFIRM ACTION is touched, the schedule for the selected DE is displayed in the graphics area. There are 25 entries on the schedule, of which 20 are changeable. The first five schedule entries are displayed in blue and represent monitored values which cannot be altered by the operator. The remaining 20 values are displayed in cyan and may be modified by the operator. After the schedule has been placed on the screen, two messages appear in the text area; "-Touch Desired Value to Be Changed", "-Or, Touch CONFIRM ACTION When All Values Appear As Desired". When an operator touches a value it is highlighted in yellow to signify selection. The message "-Please Type In The New Value" appears in the text area. The operator then types in the new value, and presses the RETURN key. The system then responds with the message "-Touch CONFIRM ACTION to Execute." When the operator touches CONFIRM ACTION the new value is stored in the system and updated on the display. Alternately, the operator could have touched CANCEL ACTION to inhibit modifying the selected schedule parameter. The operator may modify any or all of the cyan colored values as many times as desired. Finally, when CONFIRM ACTION is touched, after the two messages "-Touch Desired Value to be Changed," "Or, Touch CONFIRM ACTION When All Values Appear As Desired" are displayed in the text area, the message "-Command Action Completed" is written in the text area, after which the key backlights go out and the screen returns to its previous status. Alternately, the operator may touch the CANCEL ACTION function key to inhibit any modifications made to the schedule.

2.8 CHANGE OPER Key

The CHANGE OPER Key is a multipurpose function key. It stands for "Change Operation" and allows the operator to shutdown the system, request HELP, or Change Operators. When this blue key is touched, it becomes backlighted, the menu of available operations is presented in the graphics area, and the messages "-Ready To Change Operation" and "-Touch Square Beside Desired Operation" appear in the text area. Touching a square elicits the cue "-Touch CONFIRM ACTION To Execute," and the choice is highlighted in yellow. If CONFIRM ACTION is touched, then the selected operation takes place, after which the key backlights go out and the screen display returns to its previous status. Alternately, the operator may select a different operation or function (before touching CONFIRM ACTION), at which time the screen is updated to reflect the change, or use CANCEL ACTION to inhibit any change of operation. See sections 2.8.1, 2.8.2, and 2.8.3 for a description of the available operations.

2.8.1 Change Operator Operation

The Change Operator Operation is used to signal the system that an operator change is taking place and that future commands should be recorded under a new name. It also provides system security, in that levels of operation can be defined by assigning different operators to different levels. When this operation is selected, the cue "-Enter New Operator Name From Keyboard" appears in the text area. As the operator's name is typed in, it is echoed on the screen and checked against a list of known approved operators. If the name is unknown to the system, the message "-Name Not Recognized -- Command Action Cancelled" appears in the text area and the operator change is not executed. Otherwise the cue "-Enter password" appears in the graphics area. As the operator's password is typed in, it is checked to see if it matches the one approved for the operator. If it does not, the message "-Password Not Recognized, Access Denied" appears in the text area and the operator change is not executed. Otherwise, the system responds with the message "-Password Accepted. New Operator is (name)." in the text area, and the Date/Time/Operator display is updated. Use of CANCEL ACTION is not permitted during the change operator operation. Use of CONFIRM ACTION is not required and will be ignored.

2.8.2 Request HELP Operation

Selection of the Request HELP Operation causes the system to enter the on-line HELP facility. HELP provides an overview of system features and capabilities. When the HELP facility is entered, execution of the MMI control software will be temporarily suspended, the entire screen will be erased, and HELP will begin presenting a series of screen displays. Figures 1 through 12 are the HELP screens presented on the MMI device. The operator must press the RETURN key to advance the screen display to the next presentation when review of the current display is complete. When all parts of the sequence have been displayed, execution of the MMI will resume from the point at which it was stopped. The operator may prematurely stop the HELP sequence at any time by typing S RETURN. This is the same HELP sequence which is displayed when the system is first started. Use of the special touch-activated function keys is impossible during the Request HELP Operation.

2.8.3 Shutdown Operation

Selection of the Shutdown Operation causes the MMI/EMCS simulation to stop. When this operation is initiated, the MMI begins an orderly shutdown and then displays the message "-System Shutdown" when complete. The Shutdown Operation can not be cancelled once it has begun.

2.9 CONFIRM ACTION Key

The CONFIRM ACTION key is used by the operator to signal to the system that the desired command sequence has been entered and is now ready to be executed. It is colored green because green connotes "action," and this key is used to tell the system to take action. Its use has been detailed in the descriptions of the first eight function keys.

EMCS
MAN-MACHINE
INTERFACE

MASTER HELP
ROUTINE

-PRESS "RETURN" TO CONTINUE OR "S" THEN "RETURN" TO STOP

FIGURE 1

THE CRT DISPLAY

EIGHT PRIMARY COLORS ARE AVAILABLE:

RED	GREEN	BLACK	BLUE	WHITE	YELLOW	MAGENTA	CYAN

BUT, VARIOUS COMBINATIONS CAN BE USED TO YIELD MANY "COLORS". COLOR IS ALWAYS USED TO THE BEST POSSIBLE ADVANTAGE. COLORS WERE CHOSEN BOTH FOR VISIBILITY AND APPROPRIATENESS OF CONNOTATION.

-PRESS "RETURN" TO CONTINUE OR "S" THEN "RETURN" TO STOP

FIGURE 2

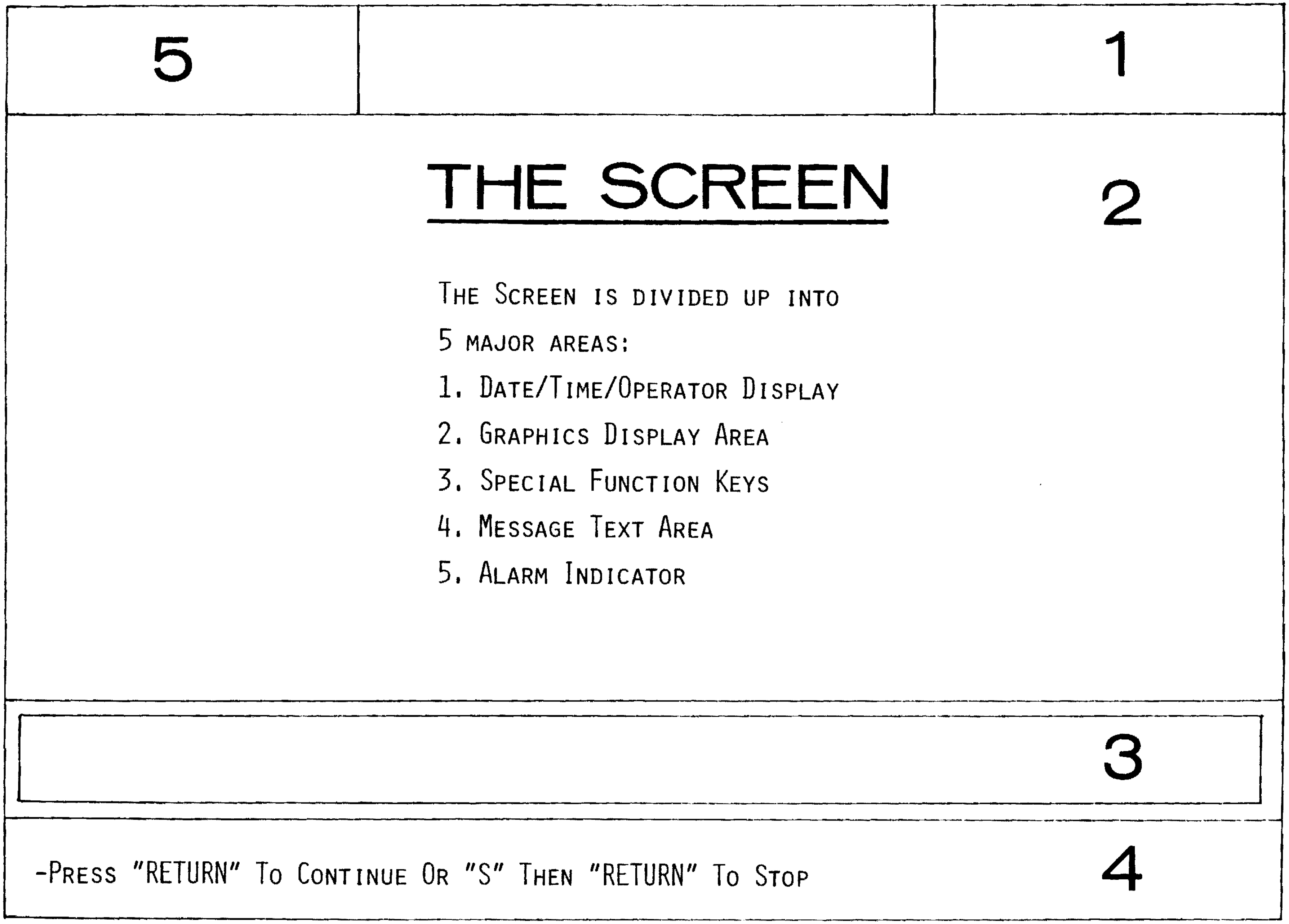

5
1
THE SCREEN
2
THE SCREEN IS DIVIDED UP INTO
5 MAJOR AREAS:
1. DATE/TIME/OPERATOR DISPLAY
2. GRAPHICS DISPLAY AREA
3. SPECIAL FUNCTION KEYS
4. MESSAGE TEXT AREA
5. ALARM INDICATOR
3
-PRESS "RETURN" TO CONTINUE OR "S" THEN "RETURN" TO STOP
4

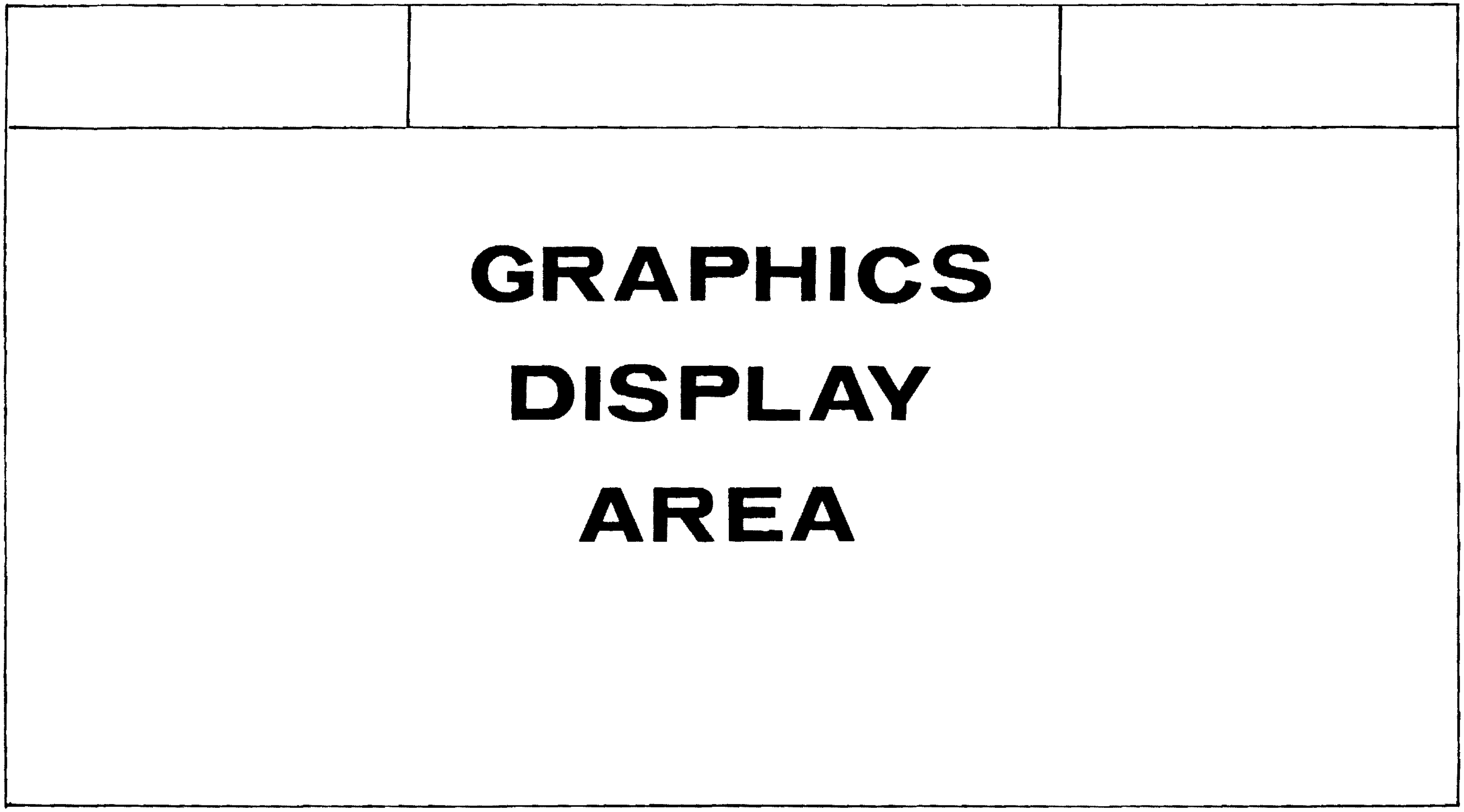
FIGURE 3

MM/DD/YY HH:MM:SS
OPERATOR NAME
TMP ###° DPT ###°

THIS AREA IS CALLED THE DATE/TIME/OPERATOR CONTINUOUS DISPLAY. IT IS USED TO SHOW THE CURRENT DATE, TIME, AND OPERATOR'S NAME, AS WELL AS THE OUTSIDE TEMPERATURE AND DEWPOINT.

-PRESS "RETURN" TO CONTINUE OR "S" THEN "RETURN" TO STOP

FIGURE 4

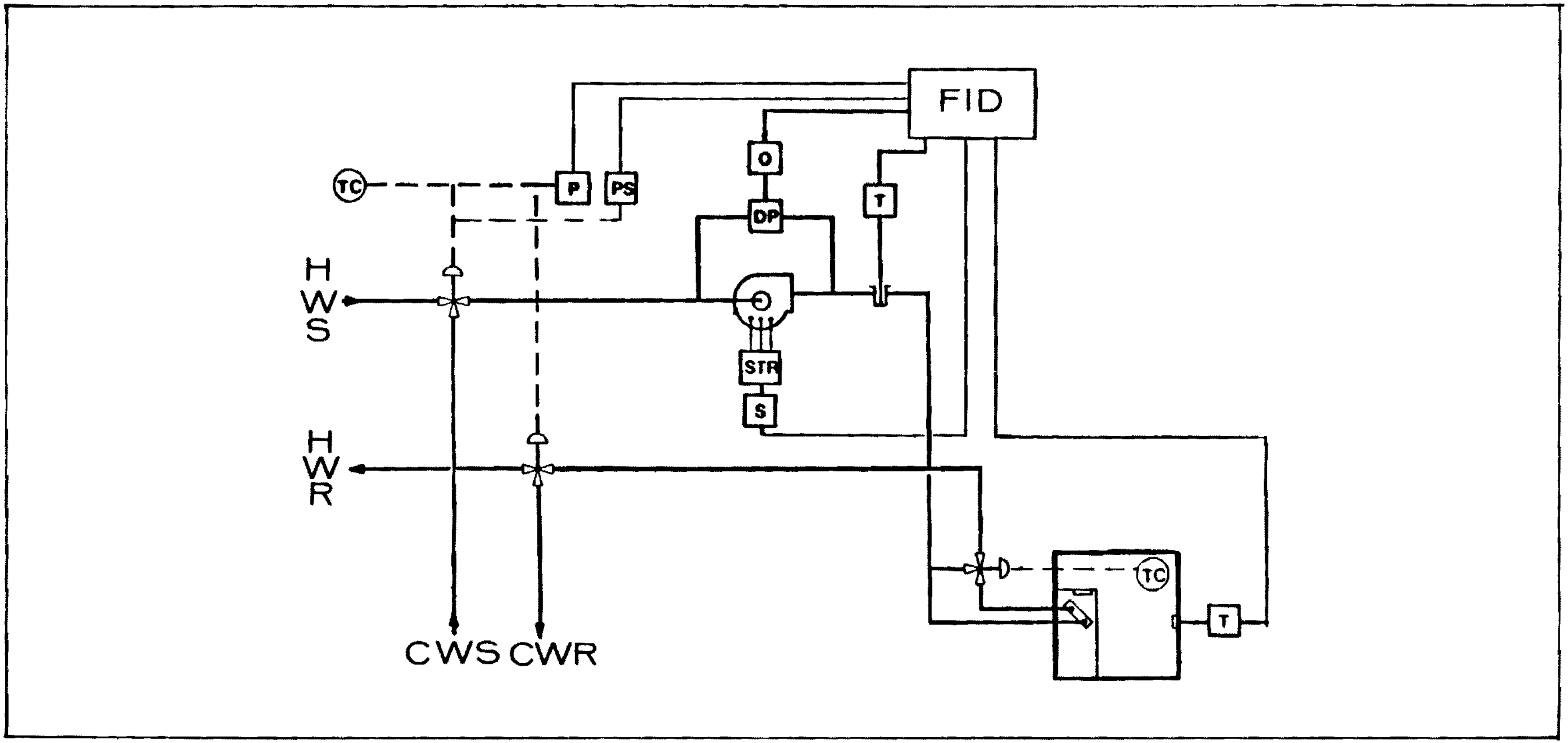


THIS AREA IS USED TO DISPLAY MULTI-COLOR GRAPHIC HVAC DIAGRAMS OF THE VARIOUS DATA ENVIRONMENTS IN THE EMCS. THE OPERATOR INTERACTS WITH THIS AREA BY TOUCHING THE DEVICE SYMBOLS AND MENU ITEMS TO BE SELECTED IN ORDER TO PERFORM A PARTICULAR COMMAND.

-PRESS "RETURN" TO CONTINUE OR "S" THEN "RETURN" TO STOP

FIGURE 5

SAMPLE DATA ENVIRONMNENT DIAGRAM



THIS IS AN EXAMPLE OF HOW A DATA ENVIRONMENT DIAGRAM APPEARS.

THE DIAGRAM IS DRAWN USING STANDARD HVAC SYMBOLOGY AND REPRESENTS A PARTICULAR AREA WHICH CAN BE CONTROLLED BY THE OPERATOR. THE OPERATOR CONTROLS THE AREA THROUGH THE USE OF THE FUNCTION KEYS IN CONJUNCTION WITH THE SELECTION OF THE DEVICES TO BE CONTROLLED. SELECTION OF DEVICES AND FUNCTION KEYS IS ACCOMPLISHED BY TOUCHING THE DESIRED FUNCTION KEY AND DEVICE SYMBOL.

-PRESS "RETURN" TO CONTINUE OR "S" THEN "RETURN" TO STOP

FIGURE 6

SPECIAL FUNCTION KEYS

THESE ARE THE TEN SPECIAL FUNCTION KEYS. THEY ARE THE PRIMARY MEANS BY WHICH THE OPERATOR INTERACTS WITH THE EMCS. THEY ARE COLOR CODED AND LABELED ACCORDING TO FUNCTION. THE AUTO/MANUAL KEY IS A SPECIAL TOGGLE KEY USED TO SELECT AS WELL AS DISPLAY THE CURRENT MODE OF OPERATION FOR A PARTICULAR DATA ENVIRONMENT.

WHEN NO DATA ENVIRONMENT IS BEING DISPLAYED, THE KEYS APPEAR AS FOLLOWS:

START/ ENABLE	STOP/ DISABLE	DISPLAY DIAGRAM	SET POINT/ LIMITS	AUTO	PRINT REPORT	MODIFY SCHED	CHANGE OPER	CONFIRM ACTION	CANCEL ACTION

-PRESS "RETURN" TO CONTINUE OR "S" THEN "RETURN" TO STOP

FIGURE 7

WHEN THE DISPLAYED DATA ENVIRONMENT IS IN THE MANUAL MODE OF OPERATION, THE KEYS WILL APPEAR AS FOLLOWS:

START/ ENABLE	STOP/ DISABLE	DISPLAY DIAGRAM	SET POINT/ LIMITS	MANUAL	PRINT REPORT	MODIFY SCHED	CHANGE OPER	CONFIRM ACTION	CANCEL ACTION

WHEN THE DISPLAYED DATA ENVIRONMENT IS IN THE AUTOMATIC MODE OF OPERATION, THE KEYS WILL APPEAR AS FOLLOWS:

START/ ENABLE	STOP/ DISABLE	DISPLAY DIAGRAM	SET POINT/ LIMITS	AUTO	PRINT REPORT	MODIFY SCHED	CHANGE OPER	CONFIRM ACTION	CANCEL ACTION

WHEN A KEY HAS BEEN TOUCHED, THE CHANGE OF COLOR TO YELLOW INDICATES FUNCTION SELECTION AND IS KNOWN AS BACKLIGHTING. IN THE FOLLOW EXAMPLE, THE PRINT REPORT KEY HAS BEEN BACKLIGHTED, THUS INDICATING THAT THE OPERATOR HAS SELECTED THE PRINT REPORT FUNCTION.

START/ ENABLE	STOP/ DISABLE	DISPLAY DIAGRAM	SET POINT/ LIMITS	AUTO	PRINT REPORT	MODIFY SCHED	CHANGE OPER	CONFIRM ACTION	CANCEL ACTION

-PRESS "RETURN" TO CONTINUE OR "S" THEN "RETURN" TO STOP

FIGURE 8

MESSAGE TEXT AREA

DIRECTLY BELOW THE FUNCTION KEYS IS THE MESSAGE TEXT AREA. THIS AREA CAN DISPLAY UP TO 4 LINES OF MESSAGES AT ONE TIME. EACH MESSAGE IS COLOR CODED ACCORDING TO WHETHER IT IS CLASSIFIED AS A CUE, ERROR, OR ALARM MESSAGE. A MESSAGE MAY APPEAR ON ANY ONE OF THE 4 LINES REGARDLESS OF ITS COLOR (CLASSIFICATION).

-PRESS "RETURN" TO CONTINUE OR "S" THEN "RETURN" TO STOP

-LINE ONE (CUES APPEAR IN GREEN)
-LINE TWO (ERRORS APPEAR IN RED)
-LINE THREE (ALARMS APPEAR AS WHITE TEXT ON A FLASHING RED BACKGROUND)
-LINE FOUR

FIGURE 9

ALARM

THIS IS THE ALARM INDICATOR. THE FLASHING WORD ALARM, ACCOMPANIED BY AN AUDIBLE TONE (BEEP), APPEARS ONLY WHEN AN ALARM CONDITION IS DETECTED. THE AUDIBLE TONE CAN BE TEMPORARILY SILENCED BY TOUCHING THE ALARM INDICATOR.

-PRESS "RETURN" TO CONTINUE OR "S" THEN "RETURN" TO STOP

FIGURE 10

WHEN A DATA ENVIRONMENT IS DISPLAYED, THE SYMBOLS WHICH ARE USED IN THE GRAPHIC REPRESENTATION ARE COLOR CODED. LIKEWISE, MESSAGES WHICH APPEAR IN THE TEXT WINDOW ARE ALSO COLOR CODED. THE FOLLOWING CHARTS SHOW HOW COLOR IS USED TO CONVEY INFORMATION.

THE FIRST CHART SHOWS THE DIFFERENT TYPES OF DEVICE SYMBOL COLORATION AND THE MEANING ASSOCIATED WITH EACH COLOR. THE SECOND CHART SHOWS THE COLORATION SCHEME FOR THE MESSAGES WHICH APPEAR IN THE MESSAGE TEXT AREA OF THE DISPLAY.

SYMBOL COLORATION SCHEME

MODE	NON-SELECTED	SELECTED	ALARM	ALARM & SELECTED
STOP / DISABLE				
START / ENABLE				

MESSAGE COLORATION SCHEME

MESSAGE TYPE	COLOR REPRESENTATION
CUE (PROMPT)	- THIS IS A CUE FOR ACTION
ERROR	- THIS IS AN ERROR
ALARM	- ALARM CONDITION DETECTED

- PRESS "RETURN" TO CONTINUE OR "S" THEN "RETURN" TO STOP

FIGURE 11

HELP

COMPLETE

THIS CONCLUDES THE ON-LINE INTRODUCTION TO THE
EMCS MAN-MACHINE INTERFACE

-PRESS "RETURN" TO CONTINUE OR "S" THEN "RETURN" TO STOP

FIGURE 12

2.10 CANCEL ACTION Key

The CANCEL ACTION key performs the opposite function of CONFIRM ACTION. It can be used any time prior to the actual execution of a command (i.e., before it has been confirmed) to signal to the system to abort the current command and return to the previous state. It is colored red because red connotes "stop," and this key is used to stop a current command. Its use has been detailed in the descriptions for the first eight function keys.

3.0 HVAC SIMULATION OPERATION AND MODIFICATION

The HVAC simulator in the MMI device is an extremely simple, table-driven model which provides the appearance of real-time operation of the HVAC system. The two most important elements of the simulation processor are the real time data base (RTDB) and the simulation table. The RTDB contains the characteristics of each EMCS point in the simulated HVAC system. Each time the demonstration device program is initiated, the RTDB (contained in Appendix F of the Design Manual) is loaded into the computer from disk files. During operation of the device, changes made via the "SET POINTS/LIMITS" function will be incorporated into the RTDB in memory. The changes will remain in effect until the program is once again restarted, at which time the original RTDB from the disk file will again be loaded, thereby "erasing" the changes.

The simulation table is an array of parameter values for each EMCS point/device number as a function a time. The parameter values in the simulation table (contained in Appendix A) were generated manually to provide a reasonable representation of actual HVAC operating conditions. The parameter values in the simulation cannot be changed during operation of the demonstration device program. The values may be changed in an off-line mode using the following procedure. With the program diskette in Drive 1, hold down the ESC, SHIFT and B keys until the cue "MEMORY SIZE" appears on the screen. The computer is now in the BASIC language mode. Type &HAFFF, press the RETURN key, type in DOS "LOAD SCENMOD" and press RETURN again. Type RUN and press RETURN. The scenario modification program will provide cues to the operator to select the time step and EMCS point/device number to be modified, enter the new value, and obtain a printed copy of the modified simulation table. EMCS point/device numbers may be determined by reference to Figures 13, 14 and 15.

The HVAC simulator operates in a time step mode, with a step duration of 100 seconds. At each time step, the program updates the parameter value of each EMCS point in the system by referring to the next column in the simulation table. The program next checks each EMCS point's current parameter value against the alarm limits for that point as contained in the RTDB. If the point's current parameter value is within set limits, no further action occurs. However, if the points' current parameter value is not within set

EMCS POINT/DEVICE NUMBERS

Building 106 Lower

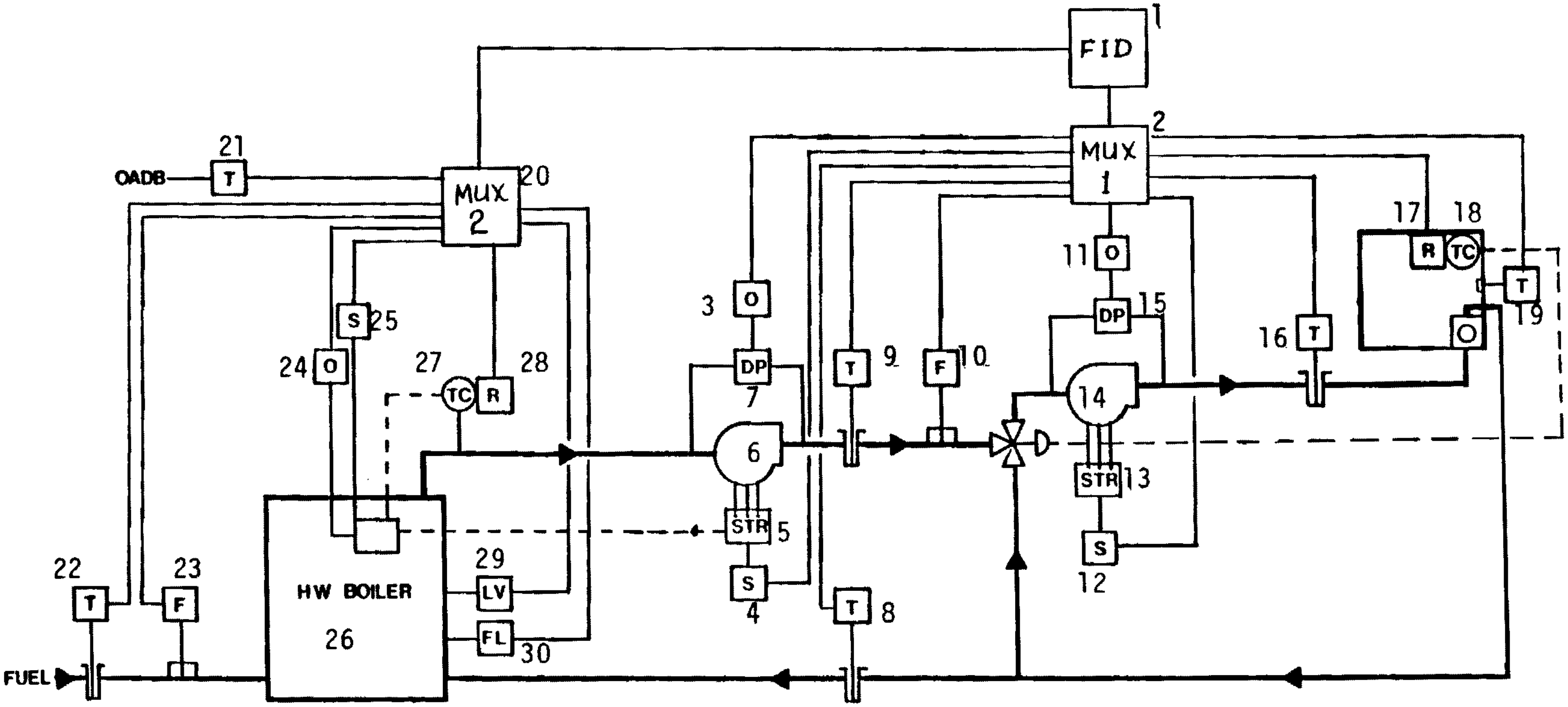


FIGURE 13

EMCS POINT/DEVICE NUMBERS
MAIN CHILLER PLANT

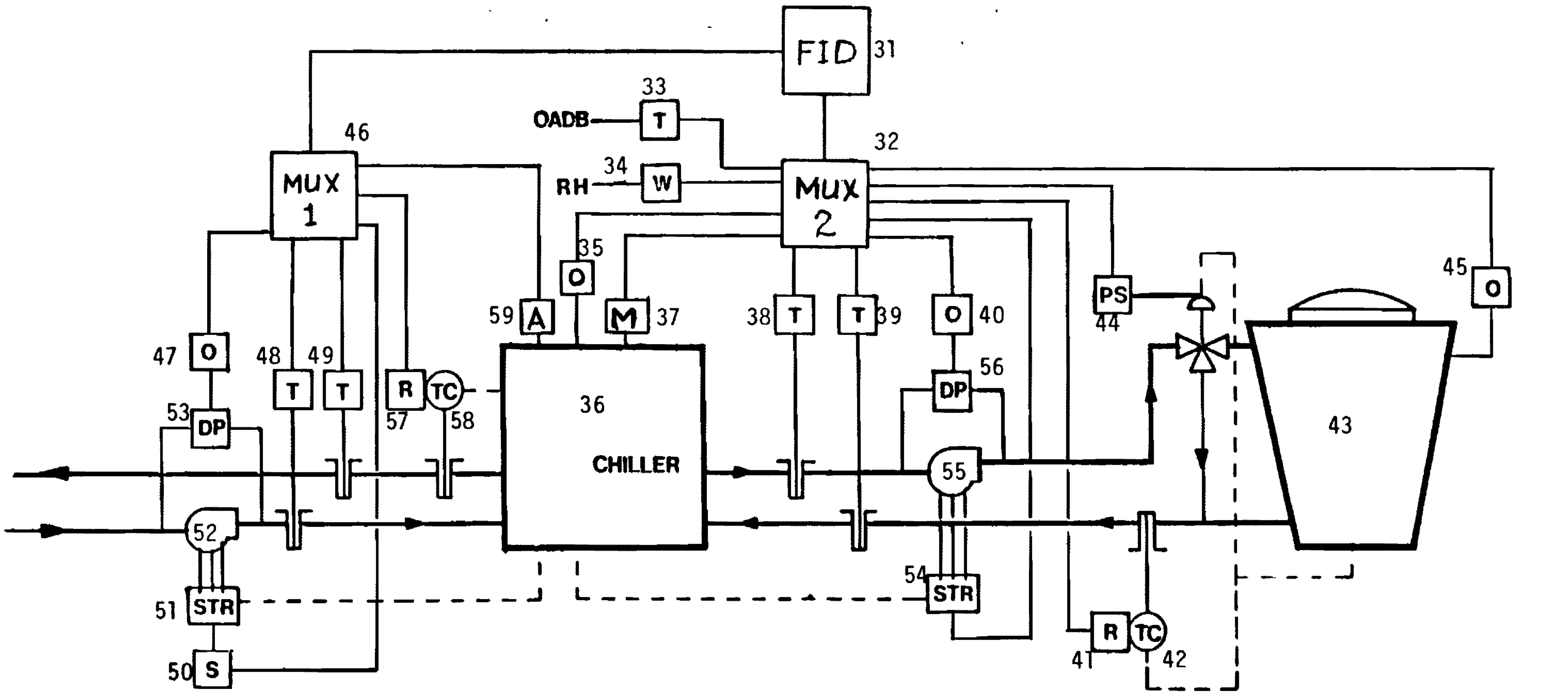


FIGURE 14

EMCS POINT/DEVICE NUMBERS
Supply No. 2

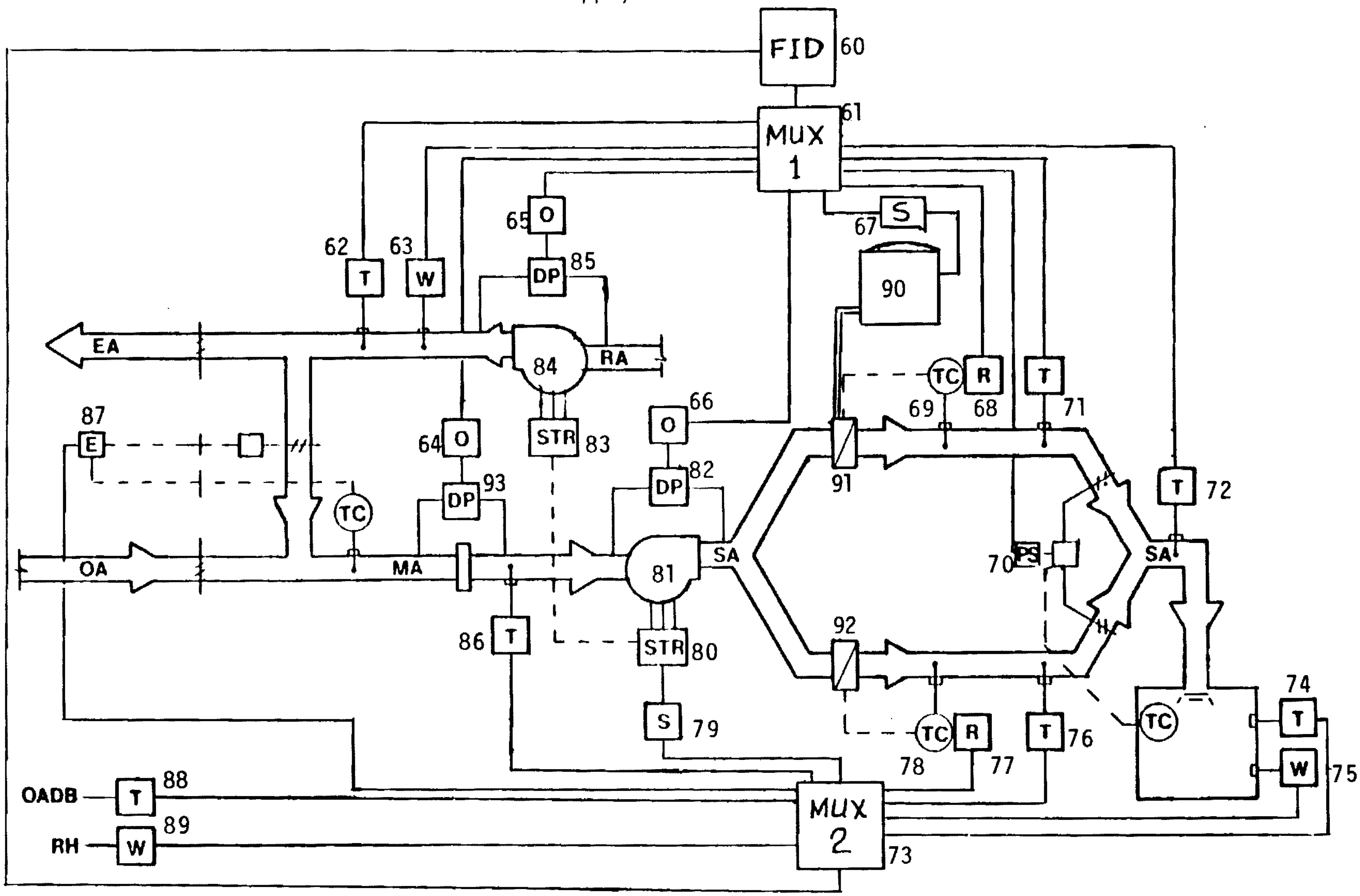


FIGURE 15

limits, an alarm condition is generated and sent back to the main program. Thus, through use of the simulation table, the HVAC model can be precisely controlled, generating alarm conditions of the type and at the time desired. More importantly, the conditions are consistent and repeatable, so that the operator can always be assured of getting the expected sequence of events. Note that the MMI program is written to accept only one EMCS point in alarm condition within a single time step, and that all points must be returned to normal conditions before a subsequent alarm condition is generated.

The time step sequence is started when the demonstration device program is initiated and will automatically loop back to the start point when the end of the table is reached. The time step sequence is temporarily halted whenever a displayed data environment (DE) diagram is put into the manual mode of operation. The time step sequence is automatically restarted from the point where it stopped whenever the displayed DE diagram is switched back to auto mode or another DE diagram is selected. Alarms generated as described above will automatically "heal" or "cure" themselves as the time step sequence moves to the next column in the simulation table. Thus, if the operator wishes to discuss a displayed alarm condition for a period exceeding the 100 second time step duration, the displayed DE diagram should be put into the manual mode to temporarily halt the time step sequence.

APPENDIX A

Simulation Table

Table Of Real-Time Device Simulation Values -- PAGE 1

DEVICE#	1	2	3	4	5	6	7
	Simulation Time Period						
1	1.0	1.0	1.0	1.0	1.0	1.0	1.0
2	1.0	1.0	1.0	1.0	1.0	1.0	1.0
3	1.0	1.0	1.0	1.0	1.0	1.0	1.0
4	1.0	1.0	1.0	1.0	1.0	1.0	1.0
5	1.0	1.0	1.0	1.0	1.0	1.0	1.0
6	1.0	1.0	1.0	1.0	1.0	1.0	1.0
7	1.0	1.0	0.0	1.0	0.0	1.0	1.0
8	110.0	110.0	110.0	110.0	110.0	110.0	110.0
9	165.0	165.0	165.0	165.0	165.0	165.0	165.0
10	10.0	10.0	10.0	0.0	10.0	0.0	10.0
11	1.0	1.0	1.0	1.0	1.0	1.0	1.0
12	1.0	1.0	1.0	1.0	1.0	1.0	1.0
13	1.0	1.0	1.0	1.0	1.0	1.0	1.0
14	1.0	1.0	1.0	1.0	1.0	1.0	1.0
15	1.0	1.0	1.0	1.0	1.0	1.0	1.0
16	160.0	160.0	160.0	160.0	160.0	160.0	160.0
17	68.0	68.0	68.0	68.0	68.0	68.0	68.0
18	68.0	68.0	68.0	68.0	68.0	68.0	68.0
19	65.0	65.0	68.0	68.0	68.0	68.0	71.0
20	1.0	1.0	1.0	1.0	1.0	1.0	1.0
21	32.0	32.0	35.0	35.0	35.0	35.0	35.0
22	50.0	50.0	50.0	50.0	50.0	50.0	50.0
23	35.0	35.0	35.0	35.0	35.0	35.0	35.0
24	1.0	1.0	1.0	1.0	1.0	1.0	1.0

Table Of Real-Time Device Simulation Values -- PAGE 2

DEVICE#	Simulation Time Period 8	9	10	11	12	13	14
1	1.0	1.0	1.0	1.0	1.0	1.0	1.0
2	1.0	1.0	1.0	1.0	1.0	1.0	1.0
3	1.0	1.0	1.0	1.0	1.0	1.0	1.0
4	1.0	1.0	1.0	1.0	1.0	1.0	1.0
5	1.0	1.0	1.0	1.0	1.0	1.0	1.0
6	1.0	1.0	1.0	1.0	1.0	1.0	1.0
7	1.0	1.0	1.0	1.0	1.0	1.0	1.0
8	110.0	110.0	110.0	115.0	115.0	115.0	115.0
9	165.0	165.0	165.0	160.0	160.0	160.0	160.0
10	10.0	10.0	10.0	10.0	10.0	10.0	10.0
11	1.0	1.0	1.0	1.0	1.0	1.0	1.0
12	1.0	1.0	1.0	1.0	1.0	1.0	1.0
13	1.0	1.0	1.0	1.0	1.0	1.0	1.0
14	1.0	1.0	1.0	1.0	1.0	1.0	1.0
15	1.0	1.0	1.0	1.0	1.0	1.0	1.0
16	160.0	135.0	135.0	155.0	155.0	155.0	155.0
17	68.0	68.0	68.0	68.0	68.0	68.0	68.0
18	68.0	68.0	68.0	68.0	68.0	68.0	68.0
19	71.0	72.0	72.0	72.0	72.0	72.0	72.0
20	1.0	1.0	1.0	1.0	1.0	1.0	1.0
21	35.0	38.0	38.0	38.0	38.0	41.0	41.0
22	50.0	50.0	50.0	50.0	50.0	50.0	50.0
23	35.0	35.0	35.0	35.0	35.0	35.0	35.0
24	1.0	1.0	1.0	1.0	1.0	1.0	1.0

Table Of Real-Time Device Simulation Values -- PAGE 3

	Simulation Time Period						
DEVICE#	15	16	17	18	19	20	21
1	1.0	1.0	1.0	1.0	1.0	1.0	1.0
2	1.0	1.0	1.0	1.0	1.0	1.0	1.0
3	1.0	1.0	1.0	1.0	1.0	1.0	1.0
4	1.0	1.0	1.0	1.0	1.0	1.0	1.0
5	1.0	1.0	1.0	1.0	1.0	1.0	1.0
6	1.0	1.0	1.0	1.0	1.0	1.0	1.0
7	1.0	1.0	1.0	1.0	1.0	1.0	1.0
8	115.0	115.0	120.0	120.0	115.0	115.0	115.0
9	160.0	160.0	155.0	155.0	160.0	160.0	160.0
10	10.0	10.0	10.0	10.0	10.0	10.0	10.0
11	1.0	1.0	1.0	1.0	1.0	1.0	1.0
12	1.0	1.0	1.0	1.0	1.0	1.0	1.0
13	1.0	1.0	1.0	1.0	1.0	1.0	1.0
14	1.0	1.0	1.0	1.0	1.0	1.0	1.0
15	1.0	1.0	1.0	1.0	1.0	1.0	1.0
16	155.0	155.0	150.0	150.0	155.0	155.0	155.0
17	68.0	68.0	68.0	68.0	68.0	68.0	68.0
18	68.0	68.0	68.0	68.0	68.0	68.0	68.0
19	73.0	73.0	74.0	74.0	74.0	74.0	73.0
20	1.0	1.0	1.0	1.0	1.0	1.0	1.0
21	41.0	41.0	45.0	45.0	42.0	42.0	42.0
22	50.0	50.0	50.0	50.0	50.0	50.0	50.0
23	35.0	35.0	35.0	35.0	35.0	35.0	35.0
24	1.0	1.0	1.0	1.0	1.0	1.0	1.0

Table Of Real-Time Device Simulation Values -- PAGE 4

	Simulation Time Period						
DEVICE#	22	23	24	25	26	27	28
1	1.0	1.0	1.0	1.0	1.0	1.0	1.0
2	1.0	1.0	1.0	1.0	1.0	1.0	1.0
3	1.0	1.0	1.0	1.0	1.0	1.0	1.0
4	1.0	1.0	1.0	1.0	1.0	1.0	1.0
5	1.0	1.0	1.0	1.0	1.0	1.0	1.0
6	1.0	1.0	1.0	1.0	1.0	1.0	1.0
7	1.0	1.0	1.0	1.0	1.0	1.0	1.0
8	115.0	115.0	115.0	115.0	115.0	110.0	110.0
9	160.0	165.0	165.0	165.0	165.0	165.0	165.0
10	10.0	10.0	10.0	10.0	10.0	10.0	10.0
11	1.0	1.0	1.0	1.0	1.0	1.0	1.0
12	1.0	1.0	1.0	1.0	1.0	1.0	1.0
13	1.0	1.0	1.0	1.0	1.0	1.0	1.0
14	1.0	1.0	1.0	1.0	1.0	1.0	1.0
15	1.0	1.0	1.0	1.0	1.0	1.0	1.0
16	155.0	160.0	160.0	160.0	160.0	160.0	160.0
17	68.0	68.0	68.0	68.0	68.0	68.0	68.0
18	68.0	68.0	68.0	68.0	68.0	68.0	68.0
19	73.0	70.0	70.0	70.0	70.0	68.0	68.0
20	1.0	1.0	1.0	1.0	1.0	1.0	1.0
21	42.0	39.0	39.0	39.0	39.0	35.0	35.0
22	50.0	50.0	50.0	50.0	50.0	50.0	50.0
23	35.0	35.0	35.0	35.0	35.0	35.0	35.0
24	1.0	1.0	1.0	1.0	1.0	1.0	1.0

Table Of Real-Time Device Simulation Values -- PAGE 5

	Simulation Time Period						
DEVICE#	1	2	3	4	5	6	7
25	1.0	1.0	1.0	1.0	1.0	1.0	1.0
26	1.0	1.0	1.0	1.0	1.0	1.0	1.0
27	165.0	165.0	165.0	165.0	165.0	165.0	165.0
28	165.0	165.0	165.0	165.0	165.0	165.0	165.0
29	1.0	1.0	1.0	1.0	1.0	1.0	1.0
30	1.0	1.0	1.0	1.0	1.0	1.0	1.0
31	1.0	1.0	1.0	1.0	1.0	1.0	1.0
32	1.0	1.0	1.0	1.0	1.0	1.0	1.0
33	75.0	75.0	75.0	75.0	78.0	78.0	78.0
34	65.0	65.0	65.0	65.0	62.0	62.0	62.0
35	1.0	1.0	1.0	1.0	1.0	1.0	1.0
36	15.0	15.0	15.0	15.0	15.0	15.0	15.0
37	85.0	85.0	85.0	85.0	88.0	88.0	88.0
38	66.0	66.0	66.0	66.0	69.0	69.0	69.0
39	1.0	1.0	1.0	1.0	1.0	1.0	1.0
40	70.0	70.0	70.0	70.0	70.0	70.0	70.0
41	70.0	70.0	70.0	70.0	70.0	70.0	70.0
42	1.0	1.0	1.0	1.0	1.0	1.0	1.0
43	50.0	50.0	45.0	45.0	40.0	40.0	35.0
44	1.0	1.0	1.0	1.0	1.0	1.0	1.0
45	1.0	1.0	1.0	1.0	1.0	1.0	1.0
46	1.0	1.0	1.0	1.0	1.0	1.0	1.0
47	55.0	55.0	55.0	55.0	55.0	55.0	55.0
48	45.0	45.0	45.0	45.0	45.0	45.0	45.0

Table Of Real-Time Device Simulation Values -- PAGE 6

DEVICE#	Simulation Time Period 8	9	10	11	12	13	14
25	1.0	1.0	1.0	1.0	1.0	1.0	1.0
26	1.0	1.0	1.0	1.0	1.0	1.0	1.0
27	165.0	165.0	165.0	165.0	165.0	165.0	165.0
28	165.0	165.0	165.0	165.0	165.0	165.0	165.0
29	1.0	1.0	1.0	1.0	1.0	1.0	1.0
30	1.0	1.0	1.0	1.0	1.0	1.0	1.0
31	1.0	1.0	1.0	1.0	1.0	1.0	1.0
32	1.0	1.0	1.0	1.0	1.0	1.0	1.0
33	78.0	81.0	81.0	81.0	81.0	85.0	85.0
34	62.0	59.0	59.0	59.0	59.0	55.0	55.0
35	1.0	1.0	1.0	1.0	1.0	1.0	1.0
36	15.0	15.0	15.0	15.0	15.0	15.0	15.0
37	88.0	91.0	91.0	91.0	91.0	95.0	95.0
38	69.0	72.0	72.0	72.0	72.0	80.0	80.0
39	1.0	1.0	1.0	1.0	1.0	1.0	1.0
40	70.0	70.0	70.0	70.0	70.0	70.0	70.0
41	70.0	70.0	70.0	70.0	70.0	70.0	70.0
42	1.0	1.0	1.0	1.0	1.0	1.0	1.0
43	35.0	35.0	35.0	30.0	30.0	70.0	70.0
44	1.0	1.0	1.0	1.0	1.0	1.0	1.0
45	1.0	1.0	1.0	1.0	1.0	1.0	1.0
46	1.0	1.0	1.0	1.0	1.0	1.0	1.0
47	55.0	55.0	55.0	55.0	55.0	55.0	55.0
48	45.0	45.0	45.0	45.0	45.0	45.0	45.0

Table Of Real-Time Device Simulation Values -- PAGE 7

	Simulation Time Period						
DEVICE#	15	16	17	18	19	20	21
25	1.0	1.0	1.0	1.0	1.0	1.0	1.0
26	1.0	1.0	1.0	1.0	1.0	1.0	1.0
27	165.0	165.0	165.0	165.0	165.0	165.0	165.0
28	165.0	165.0	165.0	165.0	165.0	165.0	165.0
29	1.0	1.0	1.0	1.0	1.0	1.0	1.0
30	1.0	1.0	1.0	1.0	1.0	1.0	1.0
31	1.0	1.0	1.0	1.0	1.0	1.0	1.0
32	1.0	1.0	1.0	1.0	1.0	1.0	1.0
33	88.0	88.0	88.0	88.0	92.0	92.0	92.0
34	52.0	52.0	52.0	52.0	50.0	50.0	50.0
35	1.0	1.0	1.0	1.0	1.0	1.0	1.0
36	15.0	15.0	2.0	2.0	15.0	15.0	15.0
37	95.0	95.0	95.0	95.0	91.0	91.0	91.0
38	73.0	73.0	74.0	74.0	74.0	74.0	74.0
39	1.0	1.0	1.0	1.0	1.0	1.0	1.0
40	70.0	70.0	70.0	70.0	70.0	70.0	70.0
41	70.0	70.0	70.0	70.0	70.0	70.0	70.0
42	1.0	1.0	1.0	1.0	1.0	1.0	1.0
43	30.0	30.0	25.0	25.0	25.0	25.0	25.0
44	1.0	1.0	1.0	1.0	1.0	1.0	1.0
45	1.0	1.0	1.0	1.0	1.0	1.0	1.0
46	1.0	1.0	1.0	1.0	1.0	1.0	1.0
47	55.0	55.0	59.0	59.0	55.0	55.0	55.0
48	45.0	45.0	45.0	45.0	45.0	45.0	45.0

Table Of Real-Time Device Simulation Values -- PAGE 8

DEVICE#	Simulation Time Period 22	23	24	25	26	27	28
25	1.0	1.0	1.0	1.0	1.0	1.0	1.0
26	1.0	1.0	1.0	1.0	1.0	1.0	1.0
27	165.0	165.0	165.0	165.0	165.0	165.0	165.0
28	165.0	165.0	165.0	165.0	165.0	165.0	165.0
29	1.0	1.0	1.0	1.0	1.0	1.0	1.0
30	1.0	1.0	1.0	1.0	1.0	1.0	1.0
31	1.0	1.0	1.0	1.0	1.0	1.0	1.0
32	1.0	1.0	1.0	1.0	1.0	1.0	1.0
33	92.0	88.0	88.0	81.0	81.0	78.0	78.0
34	50.0	54.0	54.0	58.0	58.0	62.0	62.0
35	1.0	1.0	1.0	1.0	1.0	1.0	1.0
36	15.0	15.0	15.0	15.0	15.0	15.0	15.0
37	91.0	88.0	88.0	88.0	88.0	86.0	86.0
38	74.0	73.0	73.0	67.0	67.0	64.0	64.0
39	1.0	1.0	1.0	1.0	1.0	1.0	1.0
40	70.0	70.0	70.0	70.0	70.0	70.0	70.0
41	70.0	70.0	70.0	70.0	70.0	70.0	70.0
42	1.0	1.0	1.0	1.0	1.0	1.0	1.0
43	25.0	30.0	30.0	35.0	35.0	40.0	40.0
44	1.0	1.0	1.0	1.0	1.0	1.0	1.0
45	1.0	1.0	1.0	1.0	1.0	1.0	1.0
46	1.0	1.0	1.0	1.0	1.0	1.0	1.0
47	55.0	55.0	55.0	55.0	55.0	55.0	55.0
48	45.0	45.0	45.0	45.0	45.0	45.0	45.0

Table Of Real-Time Device Simulation Values -- PAGE 9

	Simulation Time Period						
DEVICE#	1	2	3	4	5	6	7
49	1.0	1.0	1.0	1.0	1.0	1.0	1.0
50	1.0	1.0	1.0	1.0	1.0	1.0	1.0
51	1.0	1.0	1.0	1.0	1.0	1.0	1.0
52	1.0	1.0	1.0	1.0	1.0	1.0	1.0
53	1.0	1.0	1.0	1.0	1.0	1.0	1.0
54	1.0	1.0	1.0	1.0	1.0	1.0	1.0
55	1.0	1.0	1.0	1.0	1.0	1.0	1.0
56	1.0	1.0	1.0	1.0	1.0	1.0	1.0
57	45.0	45.0	45.0	45.0	45.0	45.0	45.0
58	45.0	45.0	45.0	45.0	45.0	45.0	45.0
59	0.0	0.0	0.0	0.0	0.0	0.0	0.0
60	1.0	1.0	1.0	1.0	1.0	1.0	1.0
61	1.0	1.0	1.0	1.0	1.0	1.0	1.0
62	67.0	67.0	67.0	67.0	70.0	70.0	70.0
63	40.0	40.0	40.0	40.0	40.0	40.0	40.0
64	0.0	0.0	0.0	0.0	0.0	0.0	0.0
65	1.0	1.0	1.0	1.0	1.0	1.0	1.0
66	1.0	1.0	1.0	1.0	1.0	1.0	1.0
67	1.0	1.0	1.0	1.0	1.0	1.0	1.0
68	55.0	55.0	55.0	55.0	55.0	55.0	55.0
69	55.0	55.0	55.0	55.0	55.0	55.0	55.0
70	50.0	50.0	50.0	50.0	50.0	50.0	50.0
71	55.0	55.0	55.0	55.0	55.0	55.0	55.0
72	60.0	60.0	60.0	60.0	60.0	60.0	60.0

Table Of Real-Time Device Simulation Values -- PAGE 10

DEVICE#	8	9	10	11	12	13	14
	Simulation Time Period						
49	1.0	1.0	1.0	1.0	1.0	1.0	1.0
50	1.0	1.0	1.0	1.0	1.0	1.0	1.0
51	1.0	1.0	1.0	1.0	1.0	1.0	1.0
52	1.0	1.0	1.0	1.0	1.0	1.0	1.0
53	1.0	1.0	1.0	1.0	1.0	1.0	1.0
54	1.0	1.0	1.0	1.0	1.0	1.0	1.0
55	1.0	1.0	1.0	1.0	1.0	1.0	1.0
56	1.0	1.0	1.0	1.0	1.0	1.0	1.0
57	45.0	45.0	45.0	45.0	45.0	45.0	45.0
58	45.0	45.0	45.0	45.0	45.0	45.0	45.0
59	0.0	0.0	0.0	0.0	0.0	0.0	0.0
60	1.0	1.0	1.0	1.0	1.0	1.0	1.0
61	1.0	1.0	1.0	1.0	1.0	1.0	1.0
62	70.0	73.0	73.0	73.0	73.0	75.0	75.0
63	40.0	40.0	40.0	40.0	40.0	40.0	40.0
64	0.0	0.0	0.0	0.0	0.0	0.0	0.0
65	1.0	1.0	1.0	1.0	1.0	1.0	1.0
66	1.0	1.0	1.0	1.0	1.0	1.0	1.0
67	1.0	1.0	1.0	1.0	1.0	1.0	1.0
68	55.0	55.0	55.0	55.0	55.0	55.0	55.0
69	55.0	55.0	55.0	55.0	55.0	55.0	55.0
70	50.0	50.0	50.0	50.0	50.0	50.0	50.0
71	55.0	55.0	55.0	55.0	55.0	55.0	55.0
72	60.0	60.0	60.0	60.0	60.0	60.0	60.0

Table Of Real-Time Device Simulation Values -- PAGE 11

DEVICE#	15	16	17	18	19	20	21
			Simulation Time Period				
49	1.0	1.0	1.0	1.0	1.0	1.0	1.0
50	1.0	1.0	1.0	1.0	1.0	1.0	1.0
51	1.0	1.0	1.0	1.0	1.0	1.0	1.0
52	1.0	1.0	1.0	1.0	1.0	1.0	1.0
53	1.0	1.0	1.0	1.0	1.0	1.0	1.0
54	1.0	1.0	1.0	1.0	1.0	1.0	1.0
55	1.0	1.0	1.0	1.0	1.0	1.0	1.0
56	1.0	1.0	1.0	1.0	1.0	1.0	1.0
57	45.0	45.0	45.0	45.0	45.0	45.0	45.0
58	45.0	45.0	45.0	45.0	45.0	45.0	45.0
59	0.0	0.0	0.0	0.0	0.0	0.0	0.0
60	1.0	1.0	1.0	1.0	1.0	1.0	1.0
61	1.0	1.0	1.0	1.0	1.0	1.0	1.0
62	75.0	78.0	78.0	78.0	76.0	76.0	74.0
63	40.0	40.0	40.0	40.0	40.0	40.0	40.0
64	0.0	0.0	0.0	0.0	0.0	0.0	0.0
65	1.0	1.0	1.0	1.0	1.0	1.0	1.0
66	1.0	1.0	1.0	1.0	1.0	1.0	1.0
67	1.0	1.0	1.0	1.0	1.0	1.0	1.0
68	55.0	55.0	55.0	55.0	55.0	55.0	55.0
69	55.0	55.0	55.0	55.0	55.0	55.0	55.0
70	50.0	50.0	50.0	50.0	50.0	50.0	10.0
71	55.0	55.0	55.0	55.0	55.0	55.0	55.0
72	60.0	60.0	60.0	60.0	60.0	60.0	80.0

Table Of Real-Time Device Simulation Values -- PAGE 12

DEVICE#	22	23	24	25	26	27	28
			Simulation Time Period				
49	1.0	1.0	1.0	1.0	1.0	1.0	1.0
50	1.0	1.0	1.0	1.0	1.0	1.0	1.0
51	1.0	1.0	1.0	1.0	1.0	1.0	1.0
52	1.0	1.0	1.0	1.0	1.0	1.0	1.0
53	1.0	1.0	1.0	1.0	1.0	1.0	1.0
54	1.0	1.0	1.0	1.0	1.0	1.0	1.0
55	1.0	1.0	1.0	1.0	1.0	1.0	1.0
56	1.0	1.0	1.0	1.0	1.0	1.0	1.0
57	45.0	45.0	45.0	45.0	45.0	45.0	45.0
58	45.0	45.0	45.0	45.0	45.0	45.0	45.0
59	0.0	0.0	0.0	0.0	0.0	0.0	0.0
60	1.0	1.0	1.0	1.0	1.0	1.0	1.0
61	1.0	1.0	1.0	1.0	1.0	1.0	1.0
62	74.0	71.0	71.0	71.0	71.0	71.0	71.0
63	40.0	40.0	40.0	40.0	40.0	40.0	40.0
64	0.0	0.0	0.0	0.0	0.0	0.0	0.0
65	1.0	1.0	1.0	1.0	1.0	1.0	1.0
66	1.0	1.0	1.0	1.0	1.0	1.0	1.0
67	1.0	1.0	1.0	1.0	1.0	1.0	1.0
68	55.0	55.0	55.0	55.0	55.0	55.0	55.0
69	55.0	50.0	50.0	50.0	50.0	50.0	50.0
70	10.0	10.0	10.0	10.0	10.0	10.0	10.0
71	55.0	55.0	55.0	55.0	55.0	55.0	55.0
72	80.0	60.0	60.0	60.0	60.0	60.0	60.0

Table Of Real-Time Device Simulation Values -- PAGE 13

DEVICE#	Simulation Time Period 1	2	3	4	5	6	7
73	1.0	1.0	1.0	1.0	1.0	1.0	1.0
74	73.0	73.0	73.0	73.0	73.0	73.0	73.0
75	40.0	40.0	40.0	40.0	40.0	40.0	40.0
76	80.0	80.0	80.0	80.0	80.0	80.0	80.0
77	80.0	80.0	80.0	80.0	80.0	80.0	80.0
78	80.0	80.0	80.0	80.0	80.0	80.0	80.0
79	1.0	1.0	1.0	1.0	1.0	1.0	1.0
80	1.0	1.0	1.0	1.0	1.0	1.0	1.0
81	1.0	1.0	1.0	1.0	1.0	1.0	1.0
82	1.0	1.0	1.0	1.0	1.0	1.0	1.0
83	1.0	1.0	1.0	1.0	1.0	1.0	1.0
84	1.0	1.0	1.0	1.0	1.0	1.0	1.0
85	1.0	1.0	1.0	1.0	1.0	1.0	1.0
86	58.0	58.0	60.0	60.0	62.0	62.0	64.0
87	1.0	1.0	1.0	1.0	1.0	1.0	1.0
88	50.0	50.0	55.0	55.0	60.0	60.0	65.0
89	40.0	40.0	40.0	40.0	40.0	40.0	40.0
90	1.0	1.0	1.0	1.0	1.0	1.0	1.0
91	1.0	1.0	1.0	1.0	1.0	1.0	1.0
92	1.0	1.0	1.0	1.0	1.0	1.0	1.0
93	0.0	0.0	0.0	0.0	0.0	0.0	0.0

Table Of Real-Time Device Simulation Values -- PAGE 14

DEVICE#	Simulation Time Period 8	9	10	11	12	13	14
73	1.0	1.0	1.0	1.0	1.0	1.0	1.0
74	73.0	73.0	73.0	73.0	73.0	73.0	73.0
75	40.0	40.0	40.0	40.0	40.0	40.0	40.0
76	80.0	80.0	80.0	80.0	80.0	80.0	80.0
77	80.0	80.0	80.0	80.0	80.0	80.0	80.0
78	80.0	80.0	80.0	80.0	80.0	80.0	80.0
79	1.0	1.0	1.0	1.0	1.0	1.0	1.0
80	1.0	1.0	1.0	1.0	1.0	1.0	1.0
81	1.0	1.0	1.0	1.0	1.0	1.0	1.0
82	1.0	1.0	1.0	1.0	1.0	1.0	1.0
83	1.0	1.0	1.0	1.0	1.0	1.0	1.0
84	1.0	1.0	1.0	1.0	1.0	1.0	1.0
85	1.0	1.0	1.0	1.0	1.0	1.0	1.0
86	64.0	67.0	67.0	70.0	70.0	70.0	75.0
87	1.0	1.0	1.0	1.0	1.0	1.0	1.0
88	65.0	70.0	70.0	75.0	75.0	75.0	80.0
89	40.0	38.0	38.0	38.0	38.0	38.0	35.0
90	1.0	1.0	1.0	1.0	1.0	1.0	1.0
91	1.0	1.0	1.0	1.0	1.0	1.0	1.0
92	1.0	1.0	1.0	1.0	1.0	1.0	1.0
93	0.0	0.0	0.0	0.0	0.0	0.0	0.0

DEVICE#	15	16	17	18	19	20	21
			Simulation Time Period				
73	1.0	1.0	1.0	1.0	1.0	1.0	1.0
74	73.0	73.0	73.0	73.0	73.0	73.0	77.0
75	40.0	40.0	40.0	40.0	40.0	40.0	40.0
76	80.0	80.0	80.0	80.0	80.0	80.0	80.0
77	80.0	80.0	80.0	80.0	80.0	80.0	80.0
78	80.0	80.0	80.0	80.0	80.0	80.0	80.0
79	1.0	1.0	1.0	1.0	1.0	1.0	1.0
80	1.0	1.0	1.0	1.0	1.0	1.0	1.0
81	1.0	1.0	1.0	1.0	1.0	1.0	1.0
82	1.0	1.0	1.0	1.0	1.0	1.0	1.0
83	1.0	1.0	1.0	1.0	1.0	1.0	1.0
84	1.0	1.0	1.0	1.0	1.0	1.0	1.0
85	1.0	1.0	1.0	1.0	1.0	1.0	1.0
86	75.0	80.0	80.0	77.0	77.0	73.0	73.0
87	1.0	1.0	1.0	1.0	1.0	1.0	1.0
88	80.0	85.0	85.0	80.0	80.0	75.0	75.0
89	35.0	35.0	35.0	35.0	35.0	38.0	38.0
90	1.0	1.0	1.0	1.0	1.0	1.0	1.0
91	1.0	1.0	1.0	1.0	1.0	1.0	1.0
92	1.0	1.0	1.0	1.0	1.0	1.0	1.0
93	0.0	0.0	0.0	0.0	0.0	0.0	0.0

Table Of Real-Time Device Simulation Values -- PAGE 16

DEVICE#	Simulation Time Period 22	23	24	25	26	27	28
73	1.0	1.0	1.0	1.0	1.0	1.0	1.0
74	77.0	73.0	73.0	73.0	73.0	73.0	73.0
75	40.0	40.0	40.0	40.0	40.0	40.0	40.0
76	80.0	80.0	80.0	80.0	80.0	80.0	80.0
77	80.0	80.0	80.0	80.0	80.0	80.0	80.0
78	80.0	80.0	80.0	80.0	80.0	80.0	80.0
79	1.0	1.0	1.0	1.0	1.0	1.0	1.0
80	1.0	1.0	1.0	1.0	1.0	1.0	1.0
81	1.0	1.0	1.0	1.0	1.0	1.0	1.0
82	1.0	1.0	1.0	1.0	1.0	1.0	1.0
83	1.0	1.0	1.0	1.0	1.0	1.0	1.0
84	1.0	1.0	1.0	1.0	1.0	1.0	1.0
85	1.0	1.0	1.0	0.0	0.0	1.0	1.0
86	73.0	70.0	70.0	60.0	60.0	60.0	60.0
87	1.0	1.0	1.0	1.0	1.0	1.0	1.0
88	75.0	70.0	70.0	60.0	60.0	60.0	60.0
89	38.0	38.0	38.0	40.0	40.0	40.0	40.0
90	1.0	1.0	1.0	1.0	1.0	1.0	1.0
91	1.0	1.0	1.0	1.0	1.0	1.0	1.0
92	1.0	1.0	1.0	1.0	1.0	1.0	1.0
93	0.0	0.0	0.0	0.0	0.0	0.0	0.0

APPENDIX B

CUES, ALARMS, and ERROR MESSAGES

APPENDIX B

CUES, ALARMS, and ERROR MESSAGES

All cue, error, and alarm, messages are printed in color on the screen using the following color code scheme:

- o Cues - printed in green
- o Errors - printed in red
- o Alarms - printed in white on a red blinking background.

This Appendix follows the same color coding convention in that the three types (colors) of messages are broken into separate lists.

CUES - GREEN LIST

Cue Message	Explanation
- Auto Mode Selected For DE MDEX% = 1	If the currently displayed DE is in Manual Mode and the AUTO/MANUAL key is touched, this cue appears to notify the operator that the automatic mode has been requested for this DE. The computer will not execute the mode change command until the CONFIRM ACTION key is touched.
- Command Action Cancelled MDEX% = 2	This cue appears when the operator touches the CANCEL ACTION key while entering a command sequence. The command sequence is aborted and the screen returns to its previous state.
- Command Action Completed MDEX% = 3	After the CONFIRM ACTION key is touched, the system will attempt to execute the command. If it is successful, this message will be given.
- Device (x) Has Been Started/Enabled	This cue signals the operator that the EMCS computer has started/enabled device x, where x is a device which the operator manually commanded the system to start.
- Device (x) Has Been Stopped/ Disabled	This cue signals the operator that the EMCS computer has stopped/disabled device x, where x is a device which the operator manually commanded the system to stop.
- Enter Password From Keyboard MDEX% = 4	This cue appears during the change operator command sequence upon successful acceptance of the operator°s name. At this point, the operator°s password should be typed in using the keyboard.

CUES - GREEN LIST (continued)

Cue Message	Explanation
- Manual Mode Selected For DE MDEX% = 5	If the currently displayed DE is in automatic mode and the AUTO/MANUAL key is touched, this cue appears to notify the operator that the manual mode of operation has been requested for this DE. The computer will not execute the mode change command until the CONFIRM ACTION key is touched.
- Mode Changed On DE (x)	The operator has used the AUTO/MANUAL function to change the mode of operation for DE (x). The system is acknowledging the fact that it has processed the mode change request.
- New Device Selected MDEX% = 6	This cue appears if the operator touches a second device symbol when one has already been selected. The old choice is deselected and the new choice is highlighted.
- New Function Selected MDEX% = 7	This cue appears if the operator touches one of the first eight keys when one has already been selected. The old choice is deselected and the new choice is backlighted.
- New Menu Item Selected MDEX% = 8	This cue appears if the operator touches a second menu item when one has already been selected. The old choice is deselected and the new choice is highlighted.
- New Set Point/Limits Accepted For Point (x)	The operator has used the SET POINT/LIMITS function to change the set point and/or limits for device (x). The system is acknowledging the fact that it has processed the new values and added them to the data base.

Cue Message	Explanation
- Password Accepted. New Operator Is (x).	If the password matches the one approved for the new operator during the CHANGE OPER command, this cue appears in the text area and the Date/Time/Operator display is updated. X is the new operator°s name.
- Please Type In The New Value MDEX% = 10	This cue appears during the SET POINT/LIMITS command sequence. It instructs the operator to type in a new value for the parameter that is to be modified.
- Ready To Accept Command MDEX% = 11	This cue appears whenever the MMI is idle and waiting for the operator to enter a command sequence. The cue is not imperative, but rather it is simply used to indicate that the MMI computer is operational and ready to be used.
- Ready To Change Operation MDEX% = 12	This cue appears when the operator touches the CHANGE OPER key.
- Ready To Modify Schedule MDEX% = 14	When the MODIFY SCHED command is entered, this cue appears in the text area to inform the operator that the computer is ready to make the desired schedule changes for the current DE.
- Ready To Modify Set Point And/Or Limits MDEX% = 15	When the SET POINT/LIMITS key is touched, this cue appears in the text area to inform the operator that the computer is ready to make the desired analog limit changes to the device selected.

CUES - GREEN LIST (continued)

Cue Message	Explanation
- System Shutdown MDEX% = 16	The operator has touched the CHANGE OPER function key and requested a system shutdown. This message indicates that the system has been turned off (stopped, shutdown) by the operator.
- Touch Appropriate Device Symbol MDEX% = 17	When a function key requiring a device selection is touched prior to the device symbol, this cue appears instructing the operator to select the appropriate device.
- Touch CONFIRM ACTION To Execute MDEX% = 18	This cue appears in the text area when all of the parts of a command sequence have been entered. If CONFIRM ACTION is touched, the computer will execute the command.
- Or, Touch CONFIRM ACTION When All Values Appear As Desired MDEX% = 19	This cue appears in the text area when the MODIFY SCHED function is selected. It is used to remind the operator that the computer will not execute any changes to the schedule or to control values until the CONFIRM ACTION key is touched.
- Touch Desired Function MDEX% =20	When a device symbol is touched prior to a function, this cue appears instructing the operator to touch the desired function.
- Touch Square Beside Building/Floor Plan Desired MDEX% =21	When the DISPLAY DIAGRAM key is touched, a menu of available Data Environments appears in the graphics area. This cue instructs the operator to select which DE he wishes to have displayed.

Cue Message	Explanation
- Touch Square Beside Desired Report MDEX% = 22	When the PRINT REPORT key is touched, a menu of available reports appears on the screen. This cue appears in the text area instructing the operator to select the desired report to be printed.
- Touch Square Beside Desired Value To Be Changed MDEX% = 23	The operator has entered either the MODIFY SCHED or SET POINT/LIMITS command. To modify a value, it must be selected. This is done by touching the square beside the desired value.
- Press RETURN To Continue Or S Then RETURN To Stop MDEX% = 24	The operator is examining the HELP routine. This message is a prompt signaling that the system is waiting for a response from the operator before continuing with the next screen of HELP information. If the operator types RETURN, the next screen will be displayed. If the operator types S RETURN, then the HELP sequence will be exited.
-Touch CONFIRM ACTION to Confirm Report Selection MDEX% = 9	After a report is selected from the menu of reports this message appears in the text area. This cue instructs the operator to touch CONFIRM ACTION to have his selection executed.
-Enter New Operator Name From Keyboard MDEX% = 13	After the CHANGE OPER function key is touched and the operator requests to change the system operator, this cue appears in the text area. It is used to inform the operator that the system is ready to accept a new operator.

CUES - GREEN LIST (concluded)

Cue Message	Explanation
-Touch Square Beside Desired Operation MDEX% = 43	After the CHANGE OPER function key is touched, a menu of available operations appears in the graphics area of the screen. This cue prompts the operator to select an operation.
-Touch Square Beside Desired Report Output Device MDEX% = 47	After the operator has touched the PRINT REPORT function key and requested a Report, the system places a menu of available Report Output Devices on the screen. This cue prompts the system operator to select a device.
-Touch Square Beside Desired Schedule To Be Modified MDEX% = 48	When the MODIFY SCHED function key is touched, a menu of available schedules appears on the screen. This cue appears in the text area instructing the operator to select the desired schedule to be modified.
-Touch CONFIRM ACTION To Confirm Schedule Selection MDEX% = 49	When a Schedule selection has been made by the operator, this cue appears in the text area. The message instructs the operator to touch CONFIRM ACTION to execute the command.
-Touch Desired Value To Be Changed MDEX% = 50	After a schedule has been selected by the operator and displayed on the screen, this message will appear in the text area. This cue instructs the operator to touch a value on the schedule to be modified.
-Press RETURN To Continue... MDEX% = 54	If an operator selects the CRT Screen as the Report Output Device, the Report will appear in the graphics area of the display. This cue instructs the operator to Press the RETURN key on the keyboard after viewing the report in order for system operation to continue.

ERRORS - RED LIST

Error Message	Explanation	Corrective Action
- Command Inappropriate For Device, No Action Taken MDEX%=25	A command sequence cannot be executed because the command is not appropriate for the device. For example, a fan does not have analog values. An attempt to set analog values for the fan will yield this error message.	Enter a new command sequence.
- Device Already Started/Enabled, No Action Taken MDEX% = 26	The operator has attempted to manually stop or disable a device which is already stopped/disabled.	Select a new device or function.
- Device Already Stopped/Disabled, No Action Taken MDEX% = 27	The operator has attempted to manually stop or disable a device which is already stopped/disabled.	Select a new device
- Grievous System Error MDEX% = 28	This message indicates an internal fault of the system. It should never appear under normal circumstances.	Continue processing from point of error.
- High Limit Can Not Be Lower Than Low Limit, Please Reenter MDEX% = 29	The operator has attempted to set the high limit for an analog device lower than the current low limit using the SET POINT/LIMITS command.	Enter a value for the high limit which is greater than the current low limit, or reset the low limit to a value which is less than the new high limit.
- Incomplete Command Sequence, Please Continue MDEX% = 31	The operator has touched the CONFIRM ACTION key before the entire command sequence was entered. For example, the operator requested the STOP function when no device was selected.	The operator may now continue entering the command.

ERRORS - RED LIST (continued)

Error Message	Explanation	Corrective Action
- Low Limit Cannot Be Higher Than High Limit, Please Reenter MDEX% = 32	The operator has attempted to set the low limit for an analog device higher than the current high limit using the SET POINT/ LIMITS command.	Enter a value for the low limit which is less than the current high limit or reset the high limit to a value which is greater than the new low limit.
- Name Not Recognized, Access Denied MDEX = 33	The name typed in for the new operator is unknown to the system.	Enter a new command sequence. If the CHANGE OPER function is to be selected again, check to make sure the operator name is typed correctly before typing the carriage return key. If the error occurs again, see the level 2 operator to make certain that the operator name has been placed on the list of approved system operators.
- Password Not Recognized, Access Denied MDEX% = 34	The password entered does not match the one approved for the operator; therefore, access to the system is denied.	Enter a new command sequence. If the CHANGE OPER function is to be selected again, make certain that the password is entered correctly. If the error occurs again, see the level 2 operator to obtain the correct password.
- Set Point Not Within Range Limits, Please Reenter MDEX% = 35	The operator has attempted to change the set point to a value outside of the current range limits for a particular analog device.	Enter a value for the set point that is within the current range. Alternately, change the range.
- Typographical Error, Please Reenter The Data Item. MDEX% = 36	The operator has made a typographical error while entering data from the keyboard. Typical errors include typing a letter within a number or comma within a number.	Correct the error by retyping the correct item.

ERRORS - RED LIST (continued)

Error Message	Explanation	Corrective Action
- You Cannot Set Points/Limits For That Device--Try Another MDEX% = 37	The operator has attempted to set analog values for a device which is not analog.	Select an analog device, new function, or CANCEL ACTION.
- You Must Use DISPLAY/DIAGRAM To Get A Diagram Before You Can Use This Command MDEX% = 38	The operator has selected either the START/ENABLE, STOP/DISABLE, SET POINT/LIMITS, AUTO/MANUAL, or MODIFY SCHED function when no data environment diagram is displayed.	Touch the DISPLAY DIAGRAM key and then the desired menu item to display the DE diagram. Now reenter the desired command.
- Higher Level Is Disabled, No Action Taken. MDEX% = 30	The operator has attempted to either start or enable a device, or to obtain a reading from a device which has been logically disconnected from the system.	Enable all devices higher than selected device.
- Data Environment Must Be In Manual Mode Before This Function Can Be Selected. MDEX% = 44	The operator has attempted to either START/ENABLE, STOP/DISABLE, or change the SET POINT/LIMITS on a point on a Data Environment which is in the Automatic Mode of operation.	Put the Data environment into the Manual Mode of Operation.
- Device Is Non-Selectable; Please Try Another MDEX% = 45	The operator has touched a device on the CRT screen which is not either a Function Key, cyan bordered symbol, Menu Box, or Alarm Indicator.	Aim more carefully. HINT: Use the eraser end of a pencil. Do NOT use any metal object on the surface of the screen. (The operator may receive quite a jolt and the system may crash).
- AUTO/MANUAL Function Still Pending MDEX% = 46	The operator has not completed the command sequence. CONFIRM ACTION or CANCEL ACTION must be touched to remove the system from the suspense state.	Touch either CONFIRM ACTION or CANCEL ACTION function key.

ERRORS - RED LIST (continued)

Error Message	Explanation	Corrective Action
- Value Out of Range, Please Re-enter MDEX% = 52	During Schedule Modification the operator has entered a value for the schedule entry which is not within an acceptable range, (.i.e., a multiplier value † 9.99; a season other than [S]ummer or [W]inter; a non-standard time; or a temperature outside of design limits).	Re-enter a reasonable value for the schedule entry in the exact format as it is displayed on the screen.
- Unknown Touch Target, Please Try Again MDEX% = 53	The operator has touched an area on the screen other than one of the cyan colored schedule entries.	Aim more carefully. Use the eraser end of a pencil. As a last resort calibrate the Touch Panel.

ALARMS - WHITE ON RED FLASH LIST

Alarm Message	Explanation
- Alarm Acknowledged--Audible Tone Temporarily Silenced MDEX% = 39	This cue appears when the operator touches the alarm indicator to temporarily silence the audible alarm tone. It signifies that the system has received operator acknowledgment of the alarm.
- Alarm Condition Corrected At (x)	This message signals the operator that the alarm condition no longer exists for point x, where x is a point in the system which was previously in alarm.
- Alarm Condition Detected MDEX% = 40	When an alarm condition is passed by EMCS, the MMI alarm indicator will be activated, an audible tone will sound, and this message will appear in the text area.
- To Temporarily Silence Tone And Acknowledge Alarm, Touch The Alarm Indicator MDEX% = 41	This cue appears after the operator has been notified that the alarm condition exists. If the operator so desires, the alarm indicator may to touched to temporarily silence the audible tone.
- Touch DISPLAY DIAGRAM For More Alarm Information §DEX% = 42	This Cue appears after the operator has been notified that an alarm condition exists. If the operator touches the DISPLAY DIAGRAM key it will cause the menu of data environments to be displayed. This menu will show which area has the alarm, and the operator may then command the computer to display the alarm area for subsequent corrective action.

MAN-MACHINE INTERFACE DEVICE

SOFTWARE DESIGN MANUAL

Man-Machine Interface Device

Software Design Manual

by

R.C. Coleburn

B. S. Rice

J.S. Cathcart

B. B. Wise

of the

Command and Control Branch
Electronics and Computer Systems Laboratory
Engineering Experiment Station
Georgia Institute of Technology
Atlanta, Georgia 30332

GIT/EES Project A-3008

May 1982

Prepared for

NAVAL CIVIL ENGINEERING LABORATORY
Port Hueneme, California 93043

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

1.0 INTRODUCTION

1.1 Objective

The objective of this project is to develop a prototype man-machine interface (MMI) device which will demonstrate the concept of interactive color graphics as an operator control medium for energy monitoring and control systems (EMCS). The development of and supporting rationale for the interactive color graphics concept was described in detail in a report entitled "A Man-Machine Interface for Energy Monitoring and Control Systems," dated March 1981 (CEL CR 81.013). This concept is intended to provide a "friendly" computer interface which will allow unsophisticated users to efficiently interact with a complicated EMCS, with a minimum amount of training. The purpose of this report is to document the MMI device operating capabilities and software program design.

1.2 Basis of Design

The MMI device is intended to demonstrate the technique of interactive color graphics control of an EMCS. It is not intended to simulate the functioning of an actual EMCS. The basis for the operating capabilities of the MMI device is contained in two documents, (1) "Guide Specification for Large Energy Monitoring and Control Systems" (CEGS 13947), and (2) "Design Manual for Energy Monitoring and Control Systems" (NAVFAC DM 4.9). The guide specification lists twelve tasks to be accommodated by an EMCS operator console, including:

1. Request a display of any digital or analog point, or any group of related points in the system.
2. Startup and shutdown selected systems or devices.
3. Initiate reports.
4. Request graphic displays.
5. Modify time and event scheduling.

6. Modify analog limits.

7. Adjust setpoints of selected controllers.

8. Select manual or automatic control modes.

9. Enable and disable individual points; disabling shall take precedence over all other actions.

10. Enable and disable individual FIDs.

11. Enable and disable individual MUX or IMUX panels.

12. Point definition.

Tasks (1) through (11) were considered to be within the scope of the lowest level operator's attention, and formed the basis for the operating capabilities of the MMI device. Task (12) was felt to be the province of a higher level operator, and was not implemented directly in the MMI device.

The approach taken was to develop Heating Ventilation and Air Conditioning (HVAC) subsystem color graphic diagrams, similar to those contained in NAVFAC DM 4.9, and make them interactive through use of an infrared touch panel integrated with the CRT display. This touch panel allows the operator to send commands to the EMCS by simply touching the CRT screen with a finger, thereby reducing to a minimum the requirement to type on the keyboard and eliminating confusing computer language-oriented input commands. A subgoal was to make the EMCS itself virtually transparent, so that the operator would have the feeling of interacting directly with the HVAC system, rather than with the EMCS computer. Figures 1, 2 and 3 are examples of the HVAC graphics diagrams which are to be implemented in the MMI device.

1.3 Hardware Description

As documented in CEL CR 81.013, there are several hardware features which must be present to enable satisfactory operation of a graphics operator process-control interface. Among them are rapid-draw and polygon fill, to minimize graphics drawing time; high resolution, large size CRT display, to

BUILDING 106 LOWER

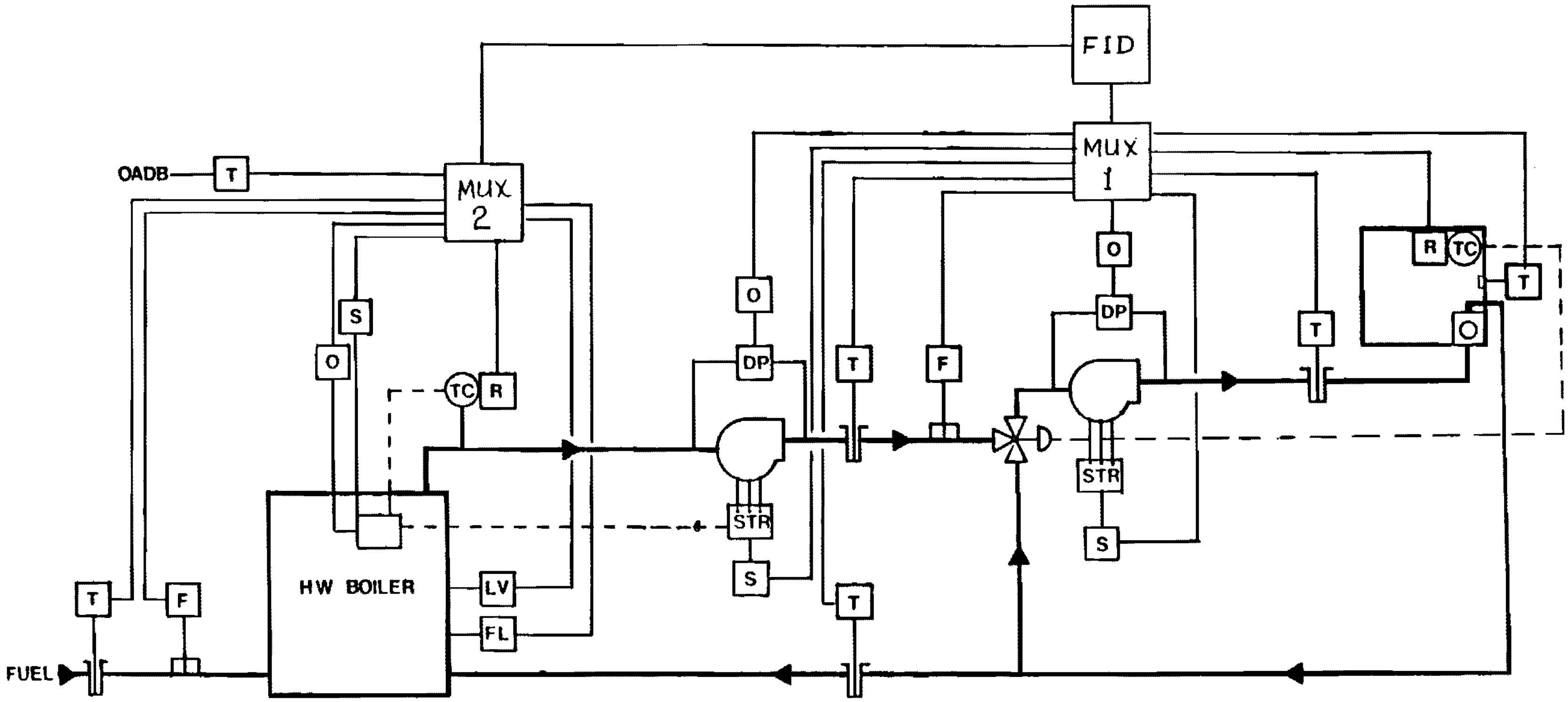


FIGURE 1

MAIN CHILLER PLANT

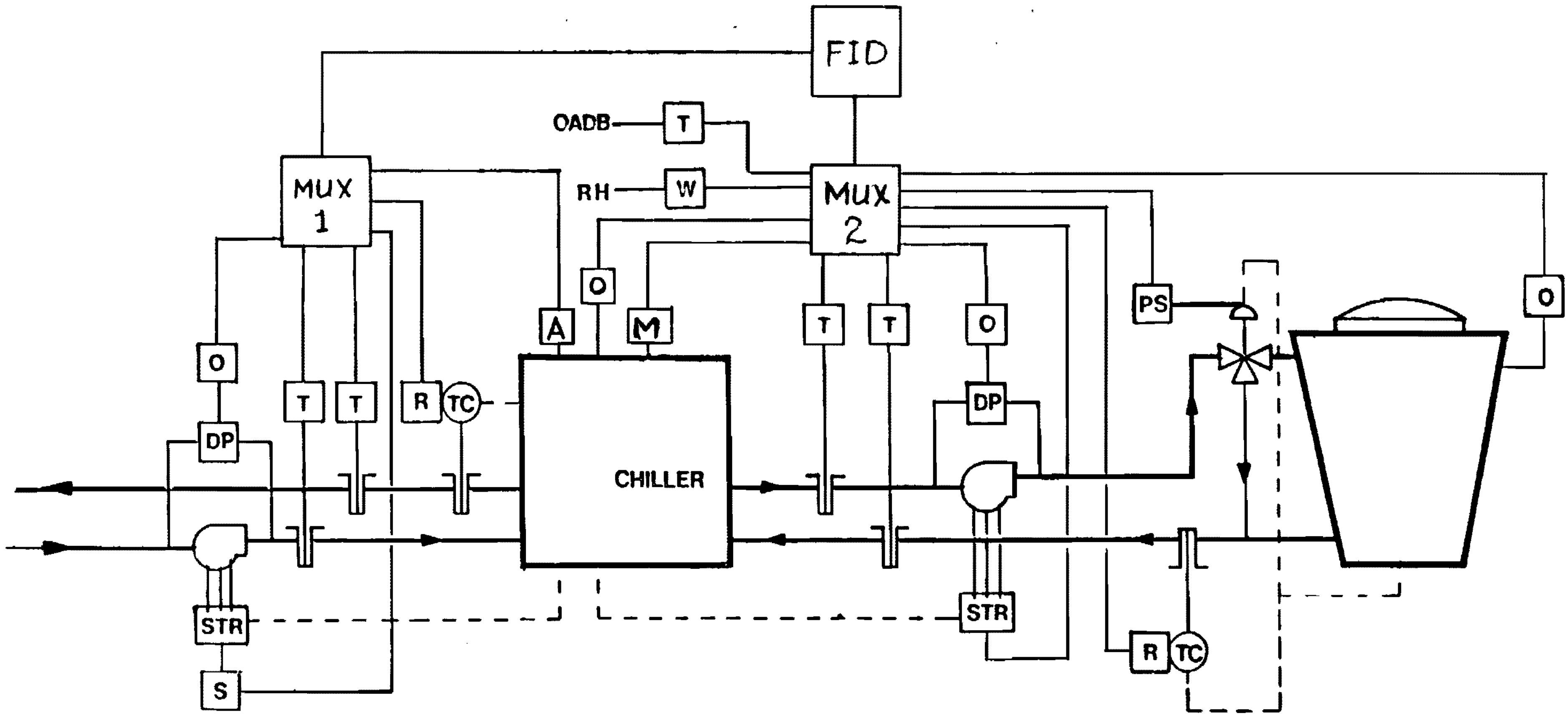


FIGURE 2

SUPPLY NO. 2

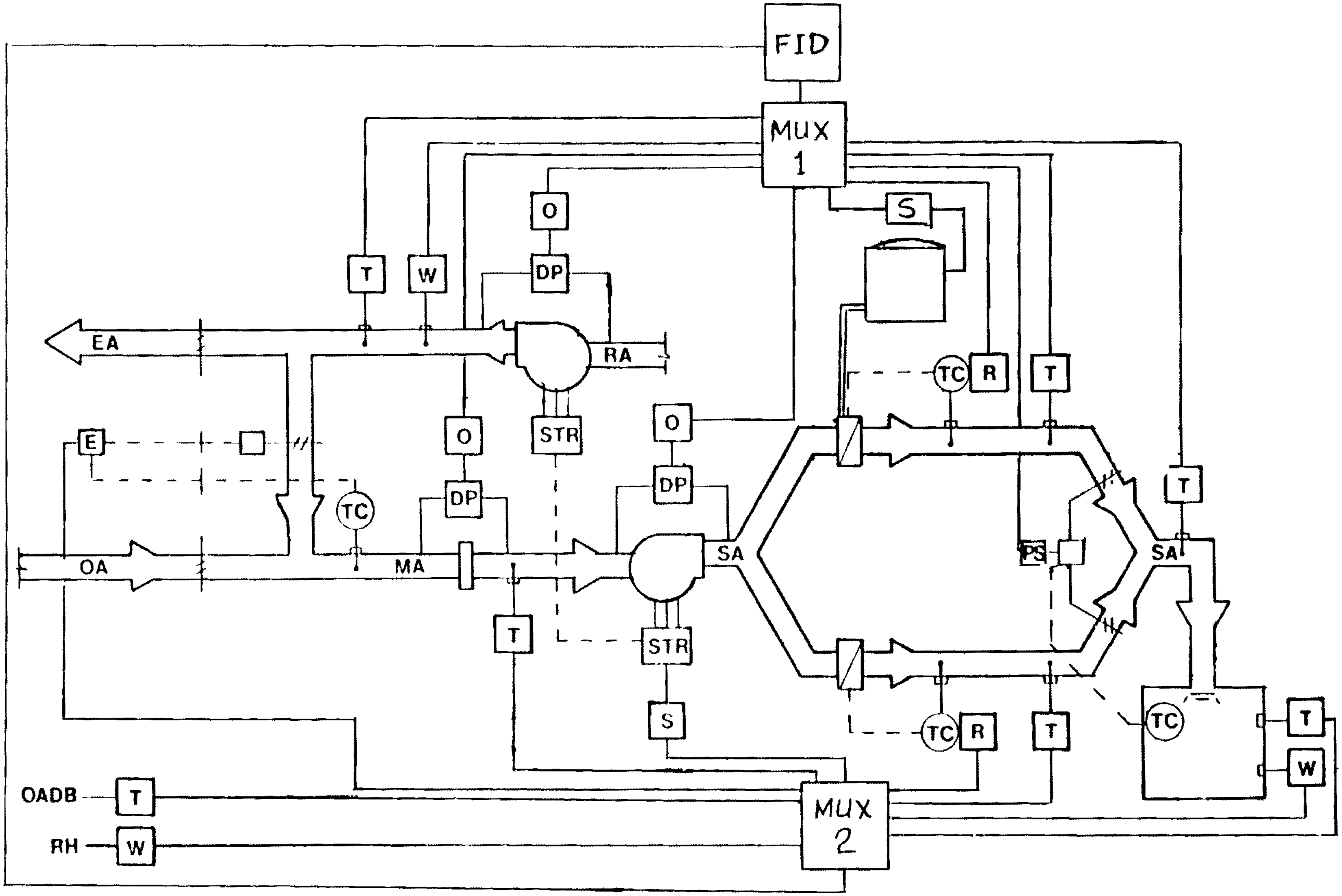
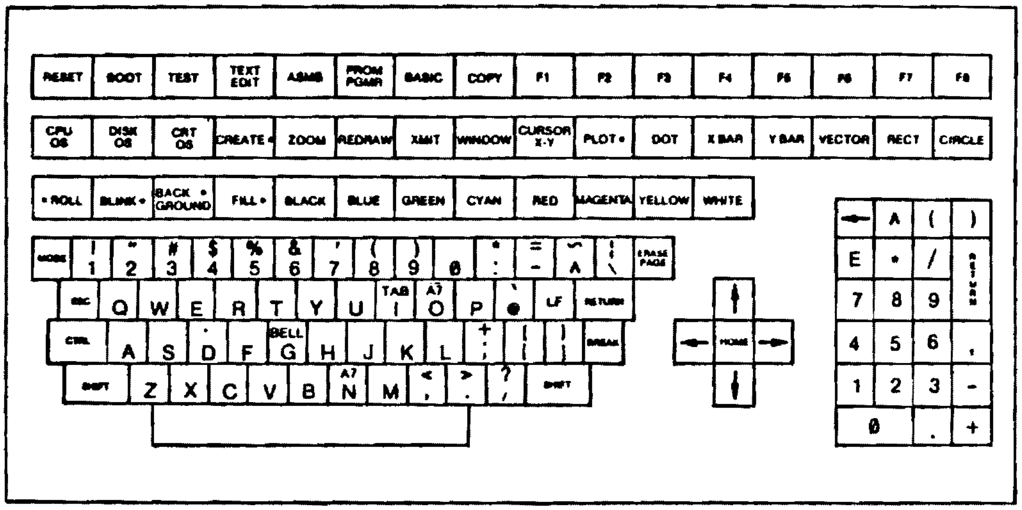


FIGURE 3

enable accurate and uncluttered reproduction of HVAC system graphic diagrams; fast CRT refresh rate, to minimize operator eyestrain and fatigue; and a positive reliable interactive touch capability, to minimize errors. It was specified that the MMI demonstration device be stand-alone in operation (i.e., that it require no host computer to perform its demonstration function). This stand-alone feature was specified to facilitate moving the MMI device between locations for conducting demonstrations.

Based upon extensive past experience with color graphics equipment, and a survey of the latest equipment readily available in the market place, the color graphics computer line produced by Chromatics, Inc. was selected for use in the MMI device implementation. The specific model chosen was the Chromatics CG3999, featuring a 19-inch, high resolution (512 x 512 pixel matrix) CRT with a 60 Hz non-interlace refresh rate. A number of available options were selected to obtain the required features enumerated above, such as complex boundary fill, extended graphics, blink, and a function key processor. In addition to these hardware color graphics capabilities, the CG3999 can operate as a stand-alone computer with its Z-80 microprocessor, 32K RAM and dual eight inch floppy disc system. To accommodate the touch panel, and to allow for system growth, three extra RS-232 interface ports were provided. While not utilized in the proposed MMI concept, a light pen was acquired to permit flexibility in color graphics system design. The keyboard accompanying the CG3999 has 128 keys (Figure 4), including numerous special function keys to activate built-in color graphics generation capabilities. This keyboard will be used to generate HVAC system graphic diagrams during the development process, and could be used by installation system programmers in the field. However, in keeping with the goal of interface simplicity and transparency for the lowest level operator, a less complex detachable typewriter-like keyboard, with a numeric key pad and minimal extra keys, will be provided for use with the MMI device (Figure 5).

Of the several touch feature technologies available in the industry, the beam interrupt type was chosen as being most durable, reliable and error-free. The IR beam interrupt touch panel from Carroll, Inc. comes in kit form and includes a printed circuit "window frame" with IR LED emitters and phototransistor diodes and its own scanning and decode logic circuits. The "window frame" is mounted around the face of the color graphics CRT and provides a grid of IR beams across the face of the tube. Resolution as small



NOTE: • INDICATES ILLUMINATED KEY.

Figure 4. 128-Key Keyboard

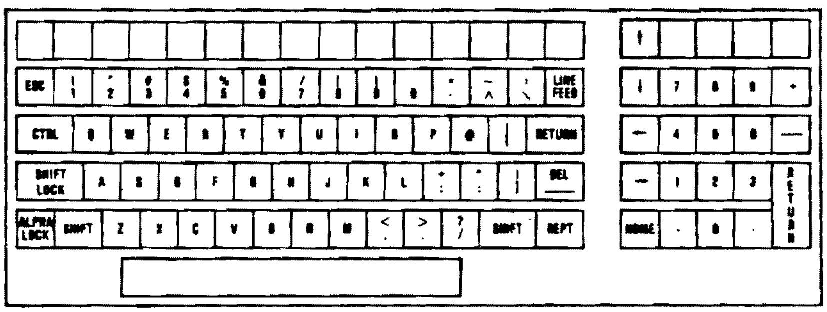


Figure 5. Simple Keyboard

as 1/8 inch can be obtained with this device. As built-in scanning circuits activate the LED's in sequence, the circuitry detects a broken IR beam resulting from an operator touch on the CRT face, and the decode logic determines the X-Y position of the touch or hit. The touch panel decode circuit connects to the Chromatics set via an RS-232 interface port at the rear of the CRT cabinet. The touch panel transmits a sequence of 3 or 4 numbers each time the touch system is activated. The first number is a unique uncover code that identifies the following two numbers as touch data. The second and third numbers represent the coordinates of the broken X and Y beams. The fourth number, which is optional, is a stop code.

To allow for hard copy output, and to demonstrate report printing capabilities, a printer has been provided. As specified, the printer provided is a NEC Model 7715 Spinwriter. This printer offers letter quality text and high resolution plotting/graphing output at 55 CPS max print rate. It attaches to the Chromatics via an RS-232 interface port. Also provided with the NEC printer is a cut sheet feeder and a vertical forms tractor. Appendix A contains a complete listing of MMI device equipment.

2.0 OPERATIONAL DESCRIPTION

In all phases of this design, special consideration has been given to the human factors associated with effectively utilizing an EMCS system. These factors are discussed in the following paragraphs, along with justification for the choice made between alternatives.

2.1 Display Screen

The screen is a CRT display unit with color graphics capability. The screen has a 512 x 512 visible dot resolution. Eight screen colors are available: red, green, blue, magneta, cyan, yellow, white, and black. The screen is divided by the software into five major areas (see Figure 6):

1. Date/Time/Operator Continuous Display
2. Graphics Display Area
3. Special Function Keys
4. Message/Text Area
5. Alarm Indicator

By assigning the types of displayed information to a consistent location on the screen, the operator will know where to look for specific information without having to scan the entire screen. Further, vital information (such as the alarm indicator) will always be present.

The CRT display unit is also equipped with an interactive touch panel utilizing infrared beam interruption technology. The touch panel allows the operator to directly interact with the graphics display, thus minimizing the use of a keyboard.

2.2 Date/Time/Operator Continuous Display

The Data/Time/Operator Continuous Display is located in the upper right corner of the screen. This area's background color is yellow, with black

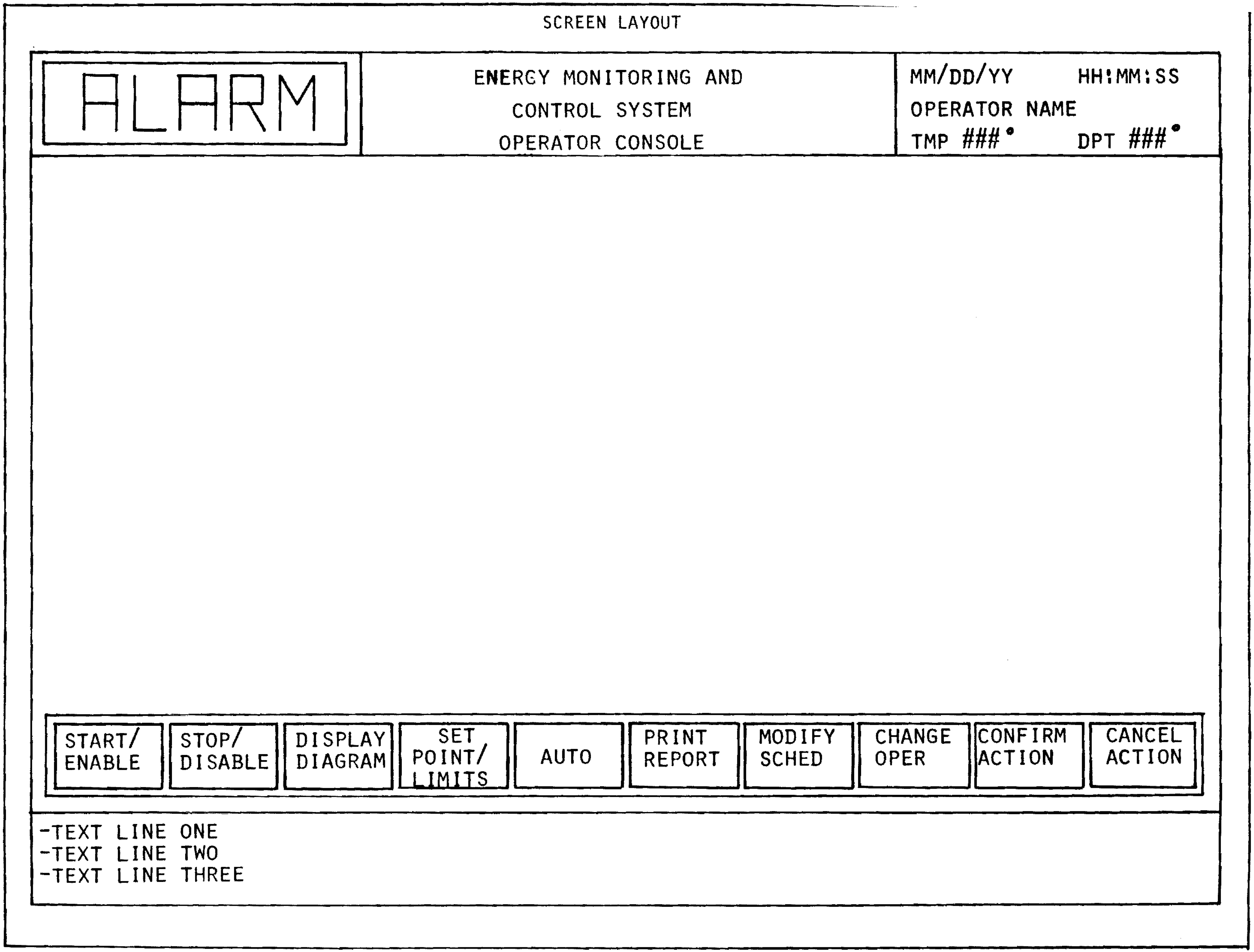

SCREEN LAYOUT
ALARM
ENERGY MONITORING AND
CONTROL SYSTEM
OPERATOR CONSOLE
MM/DD/YY HH:MM:SS
OPERATOR NAME
TMP ###° DPT ###°
START/ ENABLE
STOP/ DISABLE
DISPLAY DIAGRAM
SET POINT/ LIMITS
AUTO
PRINT REPORT
MODIFY SCHED
CHANGE OPER
CONFIRM ACTION
CANCEL ACTION
-TEXT LINE ONE
-TEXT LINE TWO
-TEXT LINE THREE

FIGURE 6

lettering used to display the current operator's name, the date, time, outside temperature, and dewpoint. Colors were chosen for optimal legibility.

2.3 Graphics Display Area

The Graphics Display Area is used for multi-color graphic display of data environment diagrams, as well as for menu selection. It occupies the middle two-thirds of the screen. Its background color is black, to provide the best background for the graphics by increasing contrast and reducing "noise." Each data environment in the system has been diagrammed using HVAC symbols and the graphics capability of the computer. The operator can command the system to display any of these diagrams for inspection or use in modifying the operation of its associated data environment. Figures 1, 2 and 3 are illustrations of sample data environment (DE) diagrams.

2.4 Special Function Keys

There are ten touch-activated Special Function Keys aligned in a row slightly below the horizontal center of the screen. These keys are the primary means by which the operator interacts with the EMCS. The keys are arranged in an order which places the most critical and frequently used functions on the ends, thus reducing operator search time. In addition, related functions are placed together, and there is a left-to-right ordering of complementary keys representing positive-to-negative connotations (e.g., START precedes STOP). In this way an operator will have less difficulty in finding keys, thus improving response time in critical situations. The touch key area is set off from the rest of the screen by a rectangular background border of cyan. The cyan background is bordered by blue lines on both the inner and outer edges. The keys themselves are located on a panel strip of black inside this cyan background border. Each key is color coded as explained in Section 2.8.

2.5 Message/Text Area

The Message/Text Area is located across the bottom of the screen. Its background color is black and all messages appear in a color chosen for good visibility and appropriate connotations. This area is used by the MMI to provide information to the operator or prompts for a response. Most of the cues which are written to this window are described in Section 2.8 through 2.8.10. This area is also used to display error messages and alarm conditions. A complete list of all cues, error messages, alarm messages, and their descriptions is given in Appendix B. The color code scheme for messages can be found in Table 1.

2.5.1 Visual Indicators

Throughout this manual, a change in color (always to yellow) indicates that a selection has been made. Use of flashing displays and text is strictly reserved for alarm conditions. Thus, whenever an operator sees something flashing on the screen, it can always be interpreted as some type of alarm indication. Likewise, whenever an item changes color to yellow, it can always be interpreted as indicating selection. Messages which appear in the text area are also color coded. Cues appear in green because green connotes "action" and a cue is a prompt signaling the operator to take action. Error messages appear in red, since red can be used to connote an abnormal condition. Alarm messages appear as white text on a flashing red background, accompanied by an audible tone. Thus, the color for alarm messages matches the alarm indicator. See Table 1 for a complete color code scheme and Table 2 for common color connotations.

2.6 Alarm Indicator

The Alarm Indicator is located in the upper left corner of the screen opposite the Date/Time/Operator window. It appears as a flashing red rectangle with the word "ALARM" printed in white. The red rectangle is bordered by a non-flashing white border. The flashing red indicator is

MESSAGE COLORATION SCHEME

Message Type	Color Representation
Cue (Prompt)	Green lettering on black background
Error Message	Red lettering on black backgroung
Alarm Message or Cue	White lettering on flashing red background

TABLE 1

COMMON COLOR CONNOTATIONS

Reds:	Stop, Abnormal Condition
Greens:	Start, Go, Action, Ready
Blues:	Calm, Normal

TABLE 2

displayed only when an alarm condition exists; however, the white border will always be present on the screen. In addition to this visual indicator, an audible tone will sound when an alarm condition is detected.

2.7 Operation of Special Function Keys

The operation of each function key follows the same general pattern, except that steps 1 and 3 may be reversed, since they are order-independent:

1. A function key is touched and becomes backlighted in yellow, indicating that it has been selected. Backlighting is defined as changing the background color of a key on the CRT screen. Only one function can be selected at a time. If another key is touched (as in step 5b), the first choice will be cancelled. Thus, the yellow backlight will always indicate which function key has been selected.

2. A cue appears in the text area instructing the operator to touch the desired device symbol or menu item (or function key if steps 1 and 3 are reversed).

3. The operator touches the desired device symbol or menu item as instructed by the cue, and the border of the device symbol (or the entire menu selection) changes color to yellow to indicate that it has been selected. This color change is referred to as highlighting. Only one device or menu item can be selected at one time, so only one device symbol or menu item can be highlighted with yellow at one time. Thus, the yellow highlighting will always indicate which device or item has been selected. If a device symbol has been selected, then the system will also display in the text area its point name, current value, and any pertinent control or alarm settings.

4. A cue appears instructing the operator to touch the CONFIRM ACTION key to execute the selected command.

5. The operator now has three choices:

    a. Touch the CONFIRM ACTION key to execute the command;

    b. Select a new function, device, or menu item, (i.e., go back to step 1 or 3.); or

    c. Touch the CANCEL ACTION key to abort the current command and return the screen to the way it was before any keys were touched.

2.7.1 Design Modifications

In the original design specifications, (CR 81.013), several keys were designed as split-function keys. Since some people have large fingers, a split-function key could pose problems in that the touch target area could be small enough to cause difficulty in selecting the desired function. Therefore, the original design has been modified to eliminate the need for split function keys. Several other modifications have been made with regard to the set of keys. Discussion of these additional modifications and justification for them is interspersed in the function key descriptions below. Keys are discussed in order from left to right on the screen.

2.7.2 Keys That Are Environment Dependent

Note that the START/ENABLE, STOP/DISABLE, SET POINT/LIMITS, and AUTO/MANUAL keys are only valid if a data environment diagram is currently being displayed. Moreover, START/ENABLE, STOP/DISABLE, and SET POINT LIMITS are valid only when the data environment is in manual mode. If one of the four keys named above is touched when no data environment is displayed, then the error message "-You Must Use DISPLAY DIAGRAM To Get A Diagram Before You Can Use This Command" appears in the text area, and then after a pause the screen appears the same as it was before any key was touched. If START/ENABLE, STOP/DISABLE, or SET POINT/LIMITS is touched when the data environment is in the automatic mode of operation, then the error message "-Data Environment must be in manual mode before this function can be selected" appears in the text area.

2.7.3 Device Types and Selection

When a data environment is displayed, many points will be represented on it. Some of these points are control points which send signals to devices in the HVAC system. Other points monitor HVAC devices and report status and alarms to the EMCS system. The remaining popints on the HVAC diagram are devices used to indicate the system's status. These three types of points are referred to as "control", "monitor (alarm)", and "indicator" points, respectively.

Furthermore, each point can also be classified as either an analog or a digital point. An analog control point operates within a range of values around a defined setpoint. An analog monitor point is used to monitor a value (e.g., temperature) and to report if that value exceeds the limits of a defined alarm range. An analog indicator point is a functional part of the HVAC system and is used by the EMCS/MMI to display on the DE that point's status (i.e., started/stopped, enabled/disabled...etc.). A digital control point is used to change the state (e.g., on/off, high/low) of an HVAC device. A digital monitor point is used to monitor the state of an HVAC device and to report when an alarm state occurs (e.g., if a device is off when it should be on). A digital indicator point is a functional part of an HVAC system and is used by the EMCS/MMI to indicate on the DE display that point's status (i.e., on/off, high/low... etc.).

When a data environment is displayed, its control and monitor points will be indicated by a cyan border and its indicator points will be displayed with a blue border. Points with cyan borders are called "selectable" points, because the operator can select such a point for various functions. Points with blue borders are "Non-selectable" points and are so called because the operator cannot interact directly with them.

The operator can select points only when a data environment containing those points is displayed. Furthermore, only one point can be selected at a time. Selectable points are either analog or digital control or monitor points and are indicated by a cyan border on the display. The operator selects a point by touching its symbol, at which time the border of the symbol will change in color from cyan to yellow to indicate selection. This is referred to as "highlighting". Any previously selected point will return to its prior unselected state. If the operator touches anything other than a control or monitor symbol (selectable device), then the error message "-Device is non-selectable; please try another" will be displayed in the text area. When a point is selected, a message will be displayed in the text area indicating the point's name; current value; setpoint and operating limits (if it is an analog control point); and alarm limits (if it is an analog monitor point).

Note that selection of a point when a data environment is in automatic mode is permitted only in order to obtain the current value of the point; no functions (enable, disable, change setpoint, etc.) can be performed upon the point. The data environment must be placed in manual mode in order for such functions to be operative. If a point is selected while the data environment is in automatic mode then the point will be highlighted, its current value and status will be displayed in the text area, and then it will immediately be deselected by the system and its border color will return to cyan.

2.7.4 Symbol Coloration

The coloration of a device symbol will always reflect its current status, according to the conventions outlined in Table 3:

Blue Border	-- Non-selectable symbol
Cyan Border	-- Selectable symbol not currently selected
Yellow Border	-- Selected symbol (only one at a time)
Green Interior	-- Device in operation (on)
Black Interior	-- Device not in operation (off)
Flashing	-- Device in alarm condition

TABLE 3

The border color of symbols is used to differentiate between selectable and non-selectable points (see Section 2.7.3). The interior color of symbols is used to differentiate between started or enabled condition (green) and a stopped or disabled condition (black).

2.8 Function Key Descriptions

In the function key descriptions which follow, it is assumed that the reader understands the general pattern of operation outlined above. Therefore, certain repetitious details will be omitted. Deviations from the pattern will of course be included. The description for the START/ENABLE key will include more detail than that of the remaining nine keys so as to allow the reader to become more familiar with the pattern.

2.8.1 START/ENABLE Key

The START/ENABLE key is used to manually start mechanical devices and enable monitoring and control devices. This key can be operated only when a data environment is in manual mode. Touching the START/ENABLE key when the data environment is in automatic mode will cause the system to display the error message, "Data environment must be in manual mode before this function can be selected." The word "device" is used to refer to any point in the system which can be operated independently, whether it be a FID, MUX, pump, fan, or data point (analog or digital input or output). In the original design, the start and stop functions were on a single key, as were the enable and disable functions. However, it was felt that placing start and enable on the same key and, analogously stop and disable on another key, would provide an easier means of control for the operator, for the following reasons:

1. Start and enable have the same connotation, differing only in that they are applied to different types of physical points, and likewise for stop and disable;
2. Start and enable are mutually exclusive functions for stop and disable;
3. The software is capable of distinguishing which devices are stopped/started and which are enabled/disabled.

In this way, the operator need not remember which devices must be started/stopped as opposed to those which must be enabled/disabled. This design also eliminates the need for split keys, as was discussed in section 2.7.1. In addition, the START/ENABLE key is colored green,which connotes "go" or "start," to further aid the operator in rapidly associating its color with its function and in scanning for the key.

The START/ENABLE key is used in the following manner. When a data environment (DE) operating in manual mode is displayed on the screen, the operator can start or enable any selectable device (indicated by a cyan border) on that DE by touching the START/ENABLE key and the appropriate device symbol on the graphic display (order is independent). When the key is touched, it changes color to yellow (hereafter referred to as backlighting the key), indicating that it has been selected. Likewise, when a device is selected, its border changes color to yellow (hereafter referred to as highlighting), indicating that it has been selected. If the key is touched first, the cue "-Touch Appropriate Device Symbol" appears in text area. If the device symbol is touched first, then the cue "-Touch Desired Function" appears in the text area. Once both a function and a device have been selected, the cue "-Touch CONFIRM ACTION To Execute" appears in the text area. At this point the operator can do one of three things:

1. Touch the CONFIRM ACTION key, at which time it becomes backlighted; the command to start/enable the device is sent to the EMCS; the START/ENABLE key backlight is extinguished; the border of the device symbol reverts back to its original color, and its interior changes to an appropriate color (see Section 2.7.3) to indicate that it has been started/enabled; the message "-Device (x) Has Been (Started/Enabled)" appears in the text area; and finally, after a slight pause (representing the time needed for the requested action to occur), the CONFIRM ACTION key backlight is extinguished.

2. Cancel the entire operation by touching the CANCEL ACTION key, at which time it becomes backlighted; the START/ENABLE key's backlight is simulaneously extinguished; the device symbol border color returns to its original color; and finally the message "-Command Action Cancelled" appears in the text area.

3. Touch a different device symbol and/or function key, at which time the appropriate symbol border color changes are made and/or the function key backlights are changed, and appropriate cues are given.

2.8.2 STOP/DISABLE Key

The STOP/DISABLE key is used to manually stop selectable mechanical devices and disable monitoring and control devices when a data environment is in the manual mode. It operates in the same manner as the START/ENABLE key. This key is colored red because red connotes "stop", and this further aids the operator in distinguishing its function. See Table 3 for an illustration of appropriate device symbol interior colors.

2.8.3 DISPLAY DIAGRAM Key

The DISPLAY DIAGRAM key is used to display HVAC diagrams of specific data environments (DE's) in the graphics display area. It is colored blue, which connotes "calmness," since its use does not directly alter the operation of the system. When the key is touched, it becomes backlighted and the building or floor selection menu is presented in the graphics area (see Figure 7). The cue "-Touch Square Beside Building/Floor Plan Desired" appears in the text area. The menu is presented in cyan and the squares are color-filled targets. When touched, the square and its text description change color to yellow to indicate which data environment has been selected, and the cue "-Touch CONFIRM ACTION To Execute" appears in the text area. If the operator touches the CONFIRM ACTION key, it becomes backlighted, the DISPLAY DIAGRAM key backlight is extinguished, the selected diagram appears in the graphics area, and finally the CONFIRM ACTION key backlight is extinguished. Alternately, the operator can change the menu or function selections, or touch the CANCEL ACTION key to cancel the command.

2.8.4 SET POINT/LIMITS Key

The SET POINT/LIMITS key is used to adjust set points and analog limits for selectable analog HVAC devices (indicated by a cyan border). This key is magenta in color. A DE diagram must be present on the screen and in manual mode when this key is touched; otherwise, the MMI will respond in a manner similiar to that described in Section 2.7.2. When this key is touched it becomes backlighted, and the cue "-Touch Appropriate Device Symbol" appears in the text area. If the symbol was touched prior to touching the key, the cue "-Touch Desired Function" would have appeared in the text area. If the function is not appropriate for the device (i.e., the device is not an analog point), the message "-You Cannot Set Points/Limits For That Device -- Please Try Another" appears in the text area. When both function and device have been selected, the cue "-Touch CONFIRM ACTION to execute" appears in the text area. At this point the operator can change the function selection, or use the CANCEL ACTION key. If the operator touches CONFIRM ACTION, a menu of choices is presented in cyan: one for the set point, one for the low limit, and another for the high limit. Each choice will have a touch-sensitive square beside it. To modify any or all points/limits, the operator merely touches the desired selections, one at a time, and the cue "-Please Type In The New Value" appears in the text area. The choice selected will always change color to yellow. The operator types in the new value and then presses

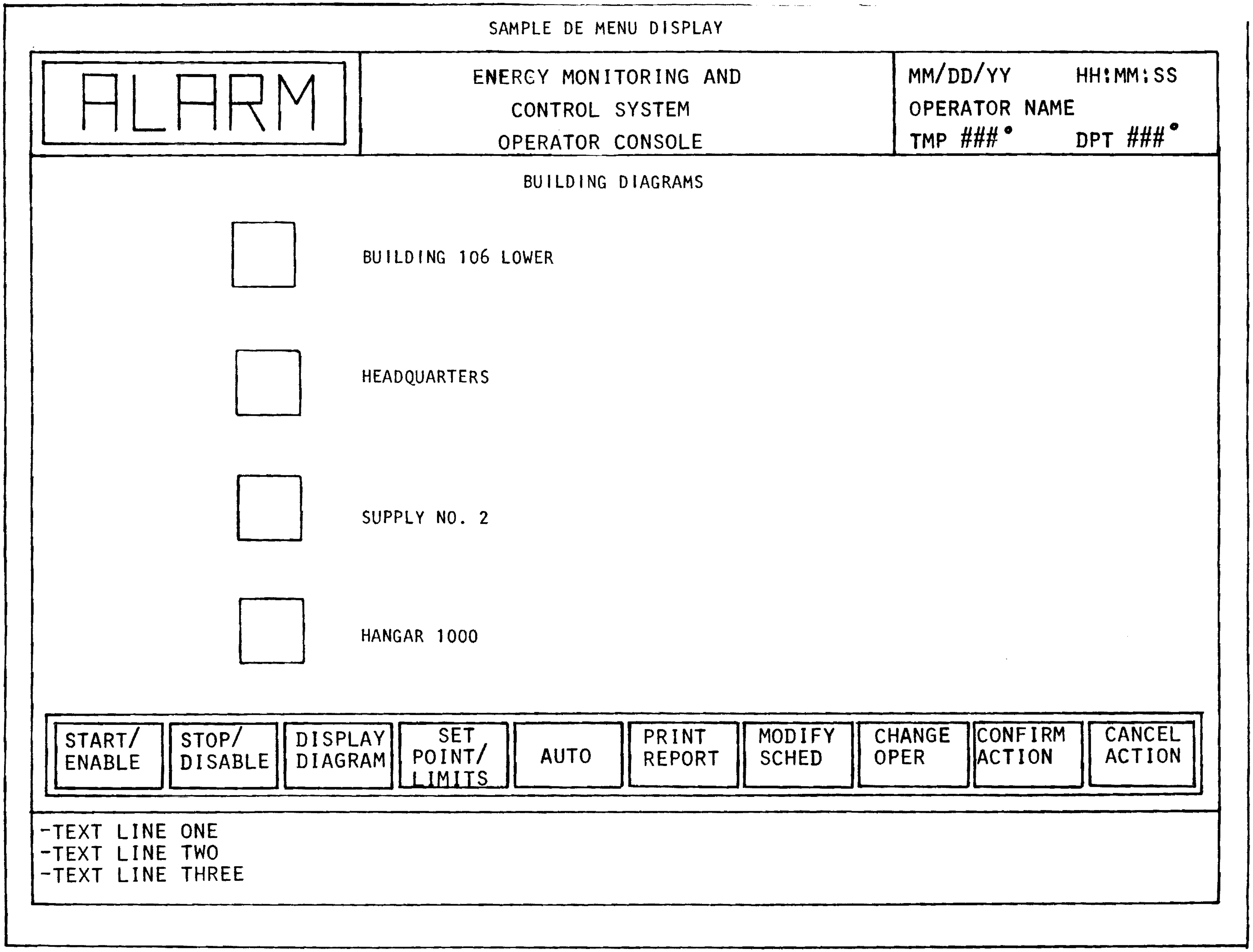

SAMPLE DE MENU DISPLAY
ALARM
ENERGY MONITORING AND
CONTROL SYSTEM
OPERATOR CONSOLE
MM/DD/YY HH:MM:SS
OPERATOR NAME
TMP ###° DPT ###°
BUILDING DIAGRAMS
BUILDING 106 LOWER
HEADQUARTERS
SUPPLY NO. 2
HANGAR 1000
START/ ENABLE
STOP/ DISABLE
DISPLAY DIAGRAM
SET POINT/ LIMITS
AUTO
PRINT REPORT
MODIFY SCHED
CHANGE OPER
CONFIRM ACTION
CANCEL ACTION
-TEXT LINE ONE
-TEXT LINE TWO
-TEXT LINE THREE

FIGURE 7

the RETURN key. At this point the operator may modify another value simply by touching its menu square and repeating the steps listed above. When all of the points/limits appear as desired, the operator merely touches CONFIRM ACTION, at which time the graphics area returns to the previous diagram, the key backlights go out, and the message "-Command Action Completed" appears in the text area. Note that the operator could still have decided not to make any changes simply by using the CANCEL ACTION key.

2.8.5 AUTO/MANUAL Key

The AUTO/MANUAL function key is a special toggle key which serves two purposes. It functions as a status indicator as to which mode the particular data environment (DE) is operating under, as well as a means of selecting the operating mode for the DE. The AUTO/MANUAL function applies only to an entire data environment, rather than to an individual device. Therefore, any device which is selected during operation of this function will be ignored. This key is colored white to make it easily distinguishable from the other function keys (no other key is white) and placed in the middle of the screen in order that it can be easily seen. The key has three different representations on the screen:

1. When no DE is displayed, the key will have the words "AUTO/MANUAL" printed on it in black letters to indicate its function.
2. When a DE is displayed which is running in automatic mode, the key will have the word "AUTO" printed on it in blue letters indicating that the DE's mode of operation is automatic. Blue lettering is used since blue connotes "calmness," and automatic mode is "calm" since it requires no intervention from the operator under normal circumstances.
3. When a DE is displayed which is running in manual mode, the key will have the word "MANUAL" printed on it in red letters indicating that the DE's mode of operation is manual. Red lettering is used because red connotes an abnormal condition, and when in manual mode the operator must take direct action to control the DE.

Touching the AUTO/MANUAL key when a DE is not displayed will generate the response indicated in Section 2.7.2, after which the screen returns to its previous state. When a DE is displayed and the key is touched, the key will backlight, and the cues "-Manual Mode Selected For DE" (or "-Auto Mode Selected For DE") and "-Touch CONFIRM ACTION To Execute" will appear in the text area. If CONFIRM ACTION is touched, the message "-Command Action Completed" will appear in the text area, the function key will toggle to indicate the new mode of operation, the backlight will extinguish, and the screen will reflect the current status of the DE. Otherwise, the operator can select a different function or use CANCEL ACTION.

2.8.6 PRINT REPORT Key

The PRINT REPORT key is used to allow the operator to initiate the display or printing of special reports. It is colored blue because blue connotes "calmness," and use of this key does not disturb the system's operation. Touching this key causes it to be backlighted and a menu of available reports, colored in cyan, to appear on the screen. The message "-Touch Square Beside Desired Report" appears in the text area. Touching a square elicits the cue "-Touch CONFIRM ACTION To Confirm Report Selection," and the choice is highlighted in yellow. If CONFIRM ACTION is touched, then the report menu screen will be erased and a menu of available output devices, colored in cyan, will appear on the screen. The message "-Touch Square Beside Desired Report Output Device" appears in the text area. Touching a square elicits the cue "-Touch CONFIRM ACTION to Execute," and the choice is highlighted in yellow. If CONFIRM ACTION is again touched, the message "-Command Action Completed" is written in the text area, after which the key backlights go out. The requested report is then sent to the printer or the CRT screen in accordance with which output device was chosen. If the report was displayed on the screen, the message "-Press RETURN To Continue..." will appear in the text area. When the RETURN key is depressed, the report will be erased and the screen will return to its previous status. If the report was printed on the printer, the screen will automatically return to its previous status. The operator may select a different report, output device, or function (before CONFIRM ACTION is touched) at which time the screen is updated to reflect the change, or use CANCEL ACTION to inhibit any report. See Figures 8, 9, and 10 for sample report formats.

ENERGY UTILIZATION SUMMARY

FRIDAY 07/31/81 10:00:32

Report On: MAIN CHILLER PLANT

TIME PERIOD	TOTAL ENERGY USAGE	MAXIMUM RATE CONSUMPTION	OUTSIDE AIR TEMP	OUTSIDE AIR HUMD
07/30/81	25 BTU	1.4 BTU	82 F	65%RH
07/31/81	23 BTU	1.2 BTU	79 F	63%RH
06/81	502 BTU	1.5 BTU	84 F	67%RH
07/81	498 BTU	1.3 BTU	80 F	64%RH

Figure 8

LOCK OUT SUMMARY

TUESDAY 04/13/82 13:04:32

Points Disabled:

BUILDING 106, Flow Meter #1 Is Disabled.

MAIN CHILLER PLANT, On/Off Status Monitor #3 Is Disabled.
MAIN CHILLER PLANT, Position Sensor #1 Is Disabled.

SUPPLY No. 2, Enthalpy Economizer Control Interface Is Disabled.
SUPPLY No. 2, Temperature Monitor #5 Is Disabled.
SUPPLY No. 2, Humidity Monitor #2 Is Disabled.

FIGURE 9

ALARM SUMMARY

Outstanding Alarms As Of: 04/13/82   13:15:40

ALARM POINT IDENTIFICATION	TIME OCCURRED	CLASS (LEVEL)	LIMITS LOW	LIMITS HIGH	CURRENT VALUE/STATUS
MESS HALL, TC2	09:45:32	2	72 F	78 F	79 F
REC BLDG 3, FAN 1	12:05:10	3	----	----	UNIT FAILURE

FIGURE 10

SCHEDULE FORMAT

	ENERGY MONITORING AND CONTROL SYSTEM OPERATOR CONSOLE	MM/DD/YY HH:MM:SS OPERATOR NAME TMP ###° DPT ###°

MAIN CHILLER PLANT - OPTIMUM START/STOP SCHEDULE

CURRENT READINGS

OUTDOOR TEMPERATURE	CALCULATED START	SPACE TEMPERATURE	WALL TEMPERATURE	SYSTEM STATUS
68.2F	0730	78.1F	89.4F	CLG

OCCUPANCY SCHEDULE

SUNDAY	MONDAY	TUESDAY	WEDNESDAY	THURSDAY	FRIDAY	SATURDAY	HOLIDAY
NA	0800	0800	0800	0800	0800	NA	NA
NA	1700	1700	1700	1700	1700	NA	NA

PARAMETER VALUES

SEASON S/W	DESIGN TEMPERATURE HEATING	DESIGN TEMPERATURE COOLING	MULTIPLIER VALUE
S	65F	80F	5.00

START/ ENABLE	STOP/ DISABLE	DISPLAY DIAGRAM	SET POINT/ LIMITS	MANUAL	PRINT REPORT	MODIFY SCHED	CHANGE OPER	CONFIRM ACTION	CANCEL ACTION

-TEXT LINE ONE
-TEXT LINE TWO
-TEXT LINE THREE

FIGURE 11

2.8.7 MODIFY SCHED Key

The MODIFY SCHED key is used to modify the automatic schedule of operation for a particular data environment (DE). Like the SET POINTS/LIMITS key, it is colored magenta to indicate that these functions are related in that they both modify stored operating parameters. This key may be used to modify the schedule for any data environment. When the MODIFY SCHED key is touched it becomes backlighted, the message "-Ready to Modify Schedule" appears in the text area, and a menu of schedule selections appears on the screen. Next, the message "-Touch Square Beside Desired Schedule To Be Modified" appears in the text area. When a schedule is selected, the message "-Touch Confirm Action to Confirm Schedule Selection" appears in the text area. At this time the operator may either select a new menu item (i.e., the schedule for another DE), touch CONFIRM ACTION, or touch CANCEL ACTION.

When CONFIRM ACTION is touched, the schedule for the selected DE is displayed in the graphics area. There are 25 entries on the schedule, of which 20 are changeable. The first five schedule entries are displayed in blue and represent monitored values which cannot be altered by the operator. The remaining 20 values are displayed in cyan and may be modified by the operator. After the schedule has been placed on the screen, two messages appear in the text area; "-Touch Desired Value to Be Changed", "-Or, Touch CONFIRM ACTION When All Values Appear As Desired". When an operator touches a value it is highlighted in yellow to signify selection. The message "-Please Type In The New Value" appears in the text area. The operator then types in the new value, and presses the RETURN key. The system then responds with the message "-Touch CONFIRM ACTION to Execute." When the operator touches CONFIRM ACTION the new value is stored in the system and updated on the display. Alternately, the operator could have touched CANCEL ACTION to inhibit modifying the selected schedule parameter. The operator may modify any or all of the cyan colored values as many times as desired. Finally, when CONFIRM ACTION is touched, after the two messages "-Touch Desired Value to be Changed," "Or, Touch CONFIRM ACTION When All Values Appear As Desired". are displayed in the text area, the message "-Command Action Completed" is written in the text area, after which the key backlights go out and the screen returns to its previous status. Alternately, the operator may touch the CANCEL ACTION function key to inhibit any modifications made to the schedule. See figure 11 for a sample schedule format.

2.8.8 CHANGE OPER Key

The CHANGE OPER Key is a multipurpose function key. It stands for "Change Operation" and allows the operator to shutdown the system, request HELP, or Change Operators. When this blue key is touched, it becomes backlighted, the menu of available operations is presented in the graphics area, and the messages "-Ready To Change Operation" and "-Touch Square Beside Desired Operation" appear in the text area. Touching a square elicits the cue "-Touch CONFIRM ACTION To Execute," and the choice is highlighted in yellow. If CONFIRM ACTION is touched, then the selected operation takes place, after which the key backlights go out and the screen display returns to its previous status. Alternately, the operator may select a different operation or function (before touching CONFIRM ACTION), at which time the screen is updated to reflect the change, or use CANCEL ACTION to inhibit any change of operation. See sections 2.8.8.1, 2.8.8.2, and 2.8.8.3 for a description of the available operations.

2.8.8.1 Change Operator Operation

The Change Operator Operation is used to signal the system that an operator change is taking place and that future commands should be recorded under a new name. It also provides system security, in that levels of operation can be defined by assigning different operators to different levels. When this operation is selected, the cue "-Enter New Operator Name From Keyboard" appears in the text area. As the operator's name is typed in, it is echoed on the screen and checked against a list of known approved operators. If the name is unknown to the system, the message "-Name Not Recognized -- Command Action Cancelled" appears in the text area and the operator change is not executed. Otherwise the cue "-Enter password" appears in the graphics area. As the operator's password is typed in, it is checked to see if it matches the one approved for the operator. If it does not, the message "-Password Not Recognized, Access Denied" appears in the text area and the operator change is not executed. Otherwise, the system responds with the message "-Password Accepted. New Operator is <u>(name)</u>." in the text area, and the Date/Time/Operator display is updated. Use of CANCEL ACTION is not permitted during the change operator operation. Use of CONFIRM ACTION is not required and will be ignored.

2.8.8.2 Request HELP Operation

Selection of the Request HELP Operation causes the system to enter the on-line HELP facility. HELP provides an overview of system features and capabilities. When the HELP facility is entered, execution of the MMI control software will be temporarily suspended, the entire screen will be erased, and HELP will begin presenting a series of screen displays. The operator must press the RETURN key to advance the screen display to the next presentation when review of the current display is complete. When all parts of the sequence have been displayed, execution of the MMI will resume from the point at which it was stopped. The operator may prematurely stop the HELP sequence at any time by typing S RETURN. This is the same HELP sequence which is displayed when the system is first started. Use of the special touch-activated function keys is impossible during the Request HELP Operation.

2.8.8.3 Shutdown Operation

Selection of the Shutdown Operation causes the MMI/EMCS simulation to stop. When this operation is initiated, the MMI begins an orderly shutdown and then displays the message "-System Shutdown" when complete. The Shutdown Operation can not be cancelled once it has begun.

2.8.9 CONFIRM ACTION Key

The CONFIRM ACTION key is used by the operator to signal to the system that the desired command sequence has been entered and is now ready to be executed. It is colored green because green connotes "action," and this key is used to tell the system to take action. Its use has been detailed in the descriptions of the first eight function keys.

2.8.10 CANCEL ACTION Key

The CANCEL ACTION key performs the opposite function of CONFIRM ACTION. It can be used any time prior to the actual execution of a command (i.e., before it has been confirmed) to signal to the system to abort the current command and return to the previous state. It is colored red because red connotes "stop," and this key is used to stop a current command. Its use has been detailed in the descriptions for the first eight function keys. The CONFIRM ACTION and CANCEL ACTION keys have been added to the original design for four major reasons:

1. To reduce operator errors,
2. To reduce the memory load on the operator,
3. To reduce operating time, and
4. To reduce operator frustration.

These objectives are accomplished by the CONFIRM and CANCEL ACTION keys by allowing for correction of errors and changes of mind before actual execution. Furthermore, they free the operator from the restriction of having to memorize and enter a fixed sequence of commands and parametric values.

2.9 Alarm Conditions

Alarms are usually triggered by some monitored value going out of bounds (e.g., temperature), or by an equipment malfunction. The MMI notifies the operator of an alarm and allows correction it in an orderly manner. The assumption behind the operational methodology described in this section is that the EMCS "stacks" alarms in a queue to be sent one at a time to the MMI on a priority assigned by the EMCS software.

2.9.1 Detection and Notification

When an alarm condition is sent by EMCS, the MMI alarm indicator will be activated and an audible tone will sound. The message "-ALARM CONDITION DETECTED" will appear flashing in red in the text area. Next, the cue "-Touch DISPLAY DIAGRAM For More Alarm Information" will appear in the text area. The audible tone will continue to sound at five-second intervals until the alarm condition is corrected, or until the operator acknowledges it. The alarm can be acknowledged by touching the alarm indicator, at which time the tone will stop sounding. This action signals to the system that the operator has acknowledged the alarm and that he will attempt corrective action. If the alarm condition still persists after two minutes have elapsed, and the operator has not attempted any corrective action during this time, the system will again begin sounding the tone at five-second intervals to remind the operator to take action.

2.9.2 Corrective Action

Following notification of the alarm, the operator can continue the current command sequence, or respond immediately to the alarm by touching the DISPLAY DIAGRAM key. Touching this key will cause the usual DE menu to appear, but with the following special indicator. The area containing the alarm will be flashing in magenta. Areas without alarms will appear in cyan and will not be flashing. The operator then selects in the usual manner (see Section 2.7.3) the alarm data environment to display. When the DE is displayed, the element or device in alarm condition will be flashing. The operator can then take the appropriate control action to correct the situation. In this way, the operator is in control and is not forced to respond immediately to an alarm and/or to follow a rigid correction algorithm.

When the operator makes an adjustment to the point in alarm, it is assumed that the alarm for that point will be disabled by the EMCS software for a period of fifteen minutes. (Five minutes is used in the demonstration system in order to facilitate the rapid presentation of the system's capabilities.) The fifteen minute alarm point disablement period is used to allow time for the operator's corrective action to become effective. If at the end of the fifteen minute disablement period the point is still in alarm, the operator will again be notified of the alarm in the manner described above. The process of notification, corrective action, and temporary alarm disablement will be repeated until the alarm condition has been corrected for the point. Note that the alarm is only disabled for the particular alarm point on which the operator attempted corrective action. All other points are still capable of generating alarms during the disablement period of a particular point. Note also that the operator will receive only one alarm at a time. This is based on the assumption that the EMCS computer will "stack" alarms in a queue to be sent to the MMI one at a time on a pre-determined priority basis. Therefore, in order to receive further alarms from EMCS after having received the first, the operator must acknowledge the first alarm by attempting some corrective action. EMCS will only send a new alarm when one of the following conditions exist:

1. There are no other active alarms or,
2. Corrective action by the operator has been attempted for all other active alarms.

During the disablement period of a particular alarm point, its device symbol will stop flashing and the disabled alarm point will appear the same as all other devices on the screen except in one respect. When the point is touched, its monitored value will be displayed in red, whereas the values of all other points will be displayed in green. In this way, the operator can visually detect any alarm points which are in the alarm disablement mode and can easily monitor their current status. By monitoring the status of an alarm point during its disablement period, the operator can determine whether or not the selected action is having a positive effect in correcting the alarm condition.

3.0 SOFTWARE DESCRIPTION

3.1 Overview

The program which will simulate the man-machine interface to an EMCS consists of an executive program and several layers of subroutines, as depicted in Figure 12. The function of the executive is to decide what is going on in the simulated real-world system (keyboard input, data base update, etc.) and to call the appropriate subroutine to respond to the system event. These subroutines in turn call other subroutines to perform specific tasks. In all, there are six levels of program modules, beginning with MMI EXEC at the highest level (Level I) and progressing downward. The levels have resulted in major part from the practice of top-down, structured program design, which stresses having a well-defined task for each module and which tends to produce easily modified and debugged software systems. Level VI (the lowest level) consists of utility subroutines which are called by modules at several different levels. Table 4 provides a brief description of the function of each of these modules, arranged alphabetically. Note that even though subroutines are referenced by line number and not by name in the BASIC language, all of the subroutines have been given mnemonic names, in order to emphasize their functions and facilitate descriptions of the system operation. For each module, a description and flowchart are provided. The page numbers in Table 4 refer to the location of the module description and flowchart in Section 3.2 This material is arranged by level, as shown on Figure 12.

Tables 5 and 6 list the elements of the two major system data bases. The MMI data base contains variables which pertain to the operation of the software system. The real-time data base contains variables whose values represent real-world system events. Appendix F is also used to list some of the more lengthy real time database values.

Due to the limitations of the Chromatics computer and the BASIC language interpreter used in the MMI demonstration device, the keyboard and touch panel cannot be used simultaneously. Consequently, the special function keys will not be available when the keyboard has been enabled for input, and vice versa. This means that in the demonstration device, use of the keyboard for input will be limited to specific instances when the system requests keyboard input.

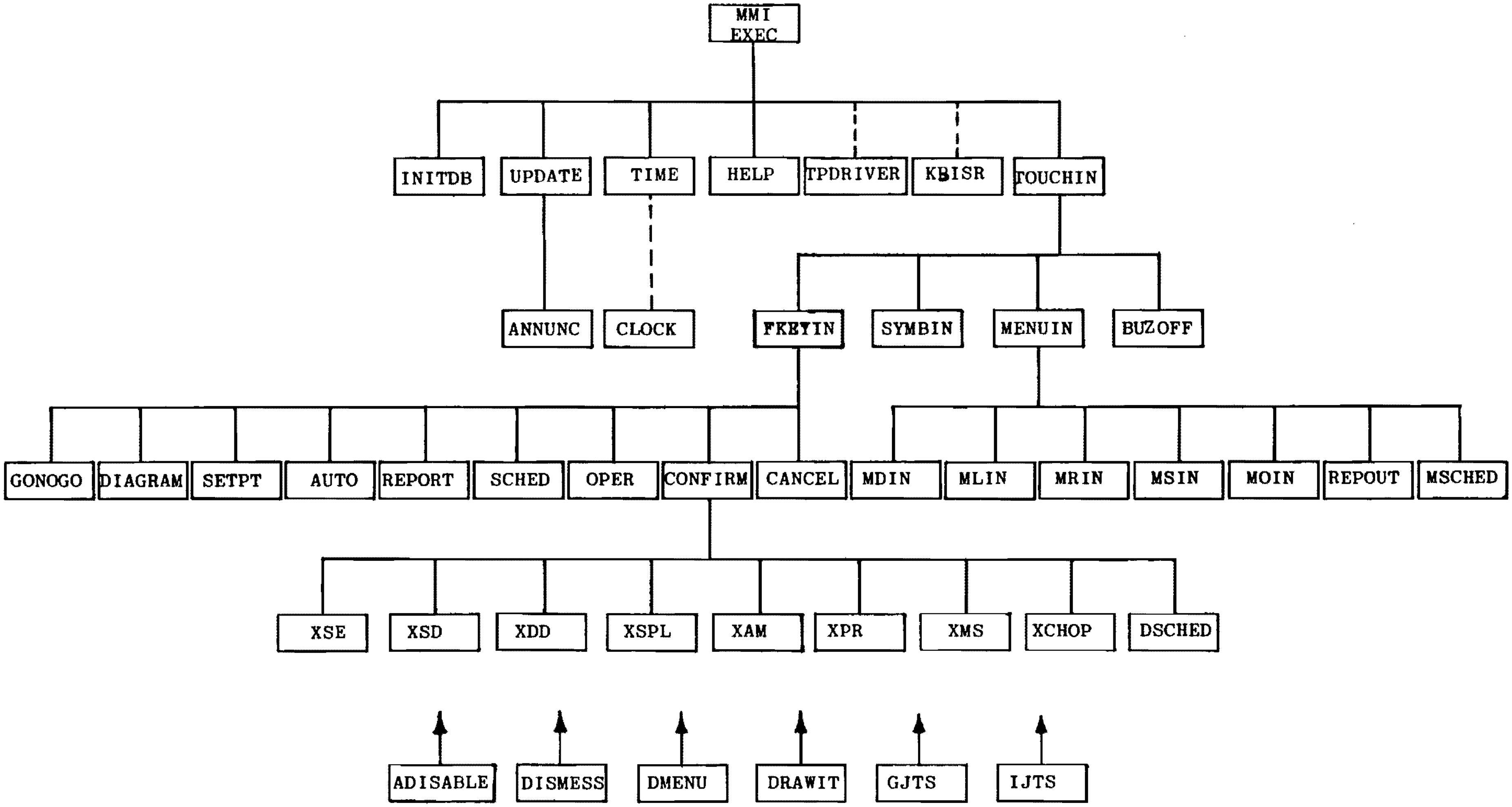


FIGURE 12. HIERARCHY OF MMI PROGRAM MODULES

The RETURN key is used to signal the end of a keyboard input response. Therefore, once keyboard input has been requested, the special function keys will not be available until the RETURN key is struck. In an actual MMI/EMCS system, these problems can be overcome through use of hardware and software with true real-time capability.

In the demonstration device, whenever a conflict arises between the selection of a special function key and a device symbol, the device symbol will always be deselected while the function key will remain selected. This feature is used to provide a standard method for resolving selection conflicts. In so doing, the assumption has been made that function selection should take precedence over device selection.

In order to facilitate the rapid presentation of the capabilities which are present in the demonstration system, time has been compressed to an approximate ratio of three minutes real time to one minute simulated time. The ratio is approximate due to the limitations of the Chromatics computer and its BASIC language interpreter. In particular, whenever the BASIC interpreter performs garbage collection (reclamation of unused, but cluttered, storage) the simulation will temporarily be halted. The garbage collection process is transparent to the user except for the noticeable delay in activity while the process is being performed. In the demonstration device, whenever the operator disables a point, that point is assumed to be logically disconnected from the system. Complete disablement means that the device and any devices it supports are disconnected, and that values and alarm status cannot be read from that device or any other device which it supports. Likewise, control commands cannot be sent to a device which has been disabled or to a device supported by a device which has been disabled. The operator must command the system to enable a point which has been disabled before it or any devices it supports can receive control commands or send values and alarm status to the system.

TABLE 4

MODULES AND THEIR FUNCTIONS

TABLE 4

MODULES AND THEIR FUNCTIONS
(continued)

TABLE 5

MMI DATA BASE

Variable	Function	Values	Set By
AQUE# (m,MQUE%)	Alarm disablement Queue	m = 0 number of point m = 1 time at which disablement period ends for that alarm.	ADISABLE
BOXLOC% (NBOXES)	Lowermost (y) coordinate of Ith menu box	(1) = 340 (3) = 180 (2) = 260	INITDB
BQUE%	Bottom of queue pointer plus 1.		MMIEXEC
C	Temporary counter		
CX%	Cursor X		XCHOP
CY%	Cursor Y		XCHOP
D%	Numerical value of date		GJTS, TIME
D1%	Argument to DRAWIT indicating type of item	-1 = Menu box 0 = Function key box 1 = Symbol 2 = Alarm indicator on 3 = Alarm indicator off	Any routine which calls DRAWIT
D2%	Argument to DRAWIT indicating type of change	1 = Change interior 2 = Change border 3 = Change interior and border	Any routine which calls DRAWIT
D3%	Argument to DRAWIT indicating color	1 = yellow border (blue or cyan otherwise) 2 = green interior (black otherwise) 4 = striped (solid otherwise) 8 = blink (non-blink otherwise)	Any routine which calls DRAWIT
	SID$ location implies 0-14 = control or monitor point (cyan border default)		

TABLE 5
(continued)

MMI DATA BASE

Variable	Function	Values	Set By
D3% (cont)	15-23 = indicator point (blue border default)		
D9%	Temporary date storage	Same as D%	IJTS
DBRDY%	Real-time data base ready flag	0 = real time data base not initialized 1 = real time data base valid	INITDB, UPDATE
DIAMEN%	Diagram or menu number currently displayed	$\geq$ 1 (see GDTYPE%) Menu: 1 = diagrams (QFCN%=3) 2 = set point/limits (analog control pt) (QFCN% = 4) 3 = set point/limits (analog monitor pt) (QFCN% = 4) 4 = reports (QFCN% = 6) 5 = operations (QFCN% = 8) 6 = devices (QFCN% = 6) 7 = schedules (QFCN% = 7) Diagram: 1-NDES% (which DE is now displayed) (i = building or DE)	DIAGRAM, SETPT, REPORT, SCHED, INITDB, CANCEL, XDD, OPER
FKLOC% (NKEYS%)	Leftmost (x) coordinate of Ith function key	(1) = 12 (6) = 257 (2) = 61 (7) = 306 (3) = 110 (8) = 355 (4) = 159 (9) = 404 (5) = 208 (10) = 453	INITDB
GDTYPE%	Graphics display type indicator	-1 = nothing displayed in graphic display area of screen 0 = menu displayed	INITDB, HELP, TOUCHIN, DMENU, CONFIRM, CANCEL,

TABLE 5
(continued)

MMI DATA BASE

Variable	Function	Values	Set By
GDTYPE% (continued)		1 = process diagram displayed	XDD,XPR
H%	Numerical Value of Hour	Current Hour of the the day in military format 0-23	GJTS,TIME
H9%	Temporary hours storage	Same as H%	IJTS
HC%	Time inhibit for help	1 = help running 0 = help not running	HELP
HTMP	Temp. storage for highest value - or alarm value operating range		DMENU
INC%	Number of seconds to increment the Julian Time leg.	UT%, 300 120, 5	UPDATE, ADISABLE, BUZOFF, MMIEXEC
JDAY%	Stores the number of days into the year on the last day of the month. NOTE: Months are represented by the indices 1 thru 12 in the 1st dimension. Leap year is indicated by a 1 in the 2nd dimension, and non-leap years are indicated by a 0 in the 2nd dimension.	(n,0) => Non-Leap year (n,1) => Leap year (1,0) = 31 (2,0) = 59 (3,0) = 90 (4,0) = 120 (5,0) = 151 (6,0) = 181 (7,0) = 212 (8,0) = 243 (9,0) = 273 (10,0) = 304 (11,0) = 334 (12,0) = 365 (1,1) = 31 (2,1) = 60 (3,1) = 91 (4,1) = 121 (5,1) = 152 (6,1) = 182 (7,1) = 213 (8,1) = 244 (9,1) = 274 (10,1) = 305 (11,1) = 335 (12,1) = 366	INITDB

TABLE 5
(continued)

MMI DATA BASE

Variable	Function	Values	Set By
KCOLR% (m,n)	Matrix which contains the foregrround and background Function key colors. NOTE: For M=1 the value in location (1,n) implies the background color of FUNCTION key number n. For M=2 the value in location (2,n) implies the color of the text for FUNCTION key number n. NOTE: When the AUTO/ MANUAL key is toggled, DRAWIT module automatically selects proper text and text color for this function key.	(1,1) = 2 (green) (2,1) = 0 (black) (1,2) = 4 (red) (2,2) = 0 (black) (1,3) = 1 (blue) (2,3) = 7 (white) (1,4) = 5 (magenta) (2,4) = 0 (black) (1,5) = 7 (white) (2,5) = 1 (blue) (for Auto Mode) (2,5) = 4 (red) (for Manual Mode) (1,6) = 1 (blue) (2,6) = 7 (white) (1,7) = 5 (magenta) (2,7) = 0 (black) (1,8) = 1 (blue) (2,8) = 0 (black) (1,9) = 2 (green) (2,9) = 0 (black) (1,10)= 4 (red) (2,10)= 0 (black)	INITDB, DRAWIT
KISR%	Keyboard input flag	0 = no input 1 = line input (CR received)	MMI EXEC, KBISR, INITDB
KTEXT$(m,n)	Matrix which contains the FUNCTION key Text Strings NOTE: M=1=> text on 1st line of function key M=2=> text on 2nd line of function key. M=3=> text on 3rd line of function key.	(1,1) = "START/" (2,1) = "" (3,1) = "ENABLE" (1,2) = "STOP/" (2,2) = "" (3,2) = "DISABLE" (1,3) = "DISPLAY" (2,3) = "" (3,3) = DIAGRAM" (1,4) = "SET" (2,4) = "POINT/" (3,4) = "LIMITS" (1,5) = "AUTO/" (2,5) = "" (3,5) = "MANUAL" (1,6) = "PRINT" (2,6) = "" (3,6) = "REPORT" (1,7) = "MODIFY" (2,7) = ""	INITDB

TABLE 5
(continued)

MMI DATA BASE

Variable	Function	Values	Set By
KTEXT$(m,n) (continued)		(3,7) = "SCHED" (1,8) = "CHANGE" (2,8) = "" (3,8) = "OPER" (1,9) = "CONFIRM" (2,9) = "" (3,9) = "ACTION" (1,10)= "CANCEL" (2,10)= "" (3,10)= "ACTION"	
LALARM%	Beginning location of monitor (alarm) point types in SID$ array.	Currently = 5	INITDB
LCNT%	Beginning location of control point types in SID$ array.	Currently = 0	INITDB
LDE%	Number of last DE displayed	1-NDES%	XDD, INITDB
LIND%	Beginning location of indicator point types in SID$ array.	Currently = 15	INITDB
LTMP	Temp. storage for lowest value - or alarm value operating range		DMENU
M%	Numerical Value of Minutes	Current number of minutes past the hour. 0-59	GJTS, TIME
M2%	Numerical Value of the day of the month	Current day of the month. 0-31	GJTS, TIME
M9%	Temporary minutes storage	Same as M%	IJTS
MDEX%	Record number of message in master file to be displayed by DISMESS	1-NMESS%	Routine which calls DISMESS
M1%	Leap year indicator flag.	0 = non-leap year 1 = leap year	TIME

TABLE 5
(continued)

MMI DATA BASE

Variable	Function	Values	Set By
MQUE%	Maximum number of alarms in queue	Currently = 5	INITDB
MT%	Numerical value of month	CURRENT MONTH 1-12	GJTS, TIME
NALARM%	Number of monitor (alarm) point types	Currently = 10	INITDB
NBOXES (DIAMEN%)	Number of menu boxes for each menu (see TT$)	(1) = 3 (4) = 3 (2) = 3 (5) = 3 (3) = 3 (6) = 2 (7) = 3	INITDB
NCNT%	Number of control point types	Currently = 5	INITDB
NHELPBF%	Number of buffer files in HELP sequence	Currently = 12	INITDB
NIND%	Number of indicator point types	Currently = 9	INITDB
NKEYS%	Number of function keys	Currently = 10	INITDB
NMESS%	Number of messages in DISMESS master file	Currently = 54	INITDB
NNSIM%	Maximum number of simulation arrays in the database.	Currently = 28	INITDB
NOPER%	Number of system operators	Currently = 5	INITDB
NUMRPTS%	Number of reports which can be printed	Currently = 3	INITDB
OLEVEL% (NOPER%)	"Security" level of Ith operator	(1) = 3 (4) = 3 (2) = 3 (5) = 3 (3) = 3	INITDB
OPER$ (NOPER%)	String containing name of Ith operator	(1) = "COLEBURN" (2) = "CATHCART" (3) = "WISE" (4) = "CANFIELD" (5) = "RICE"	INITDB

TABLE 5
(continued)

MMI DATA BASE

Variable	Function	Values	Set By
PAUSEALRM#	Pause timer for alarm tone silencer (pause for two minutes)	Julian Time at which alarm should begin sounding again.	BUZOFF
PWRD$ (NOPER%)	String containing password for Ith operator	(1) = "RANDY" (2) = "STEVE" (3) = "BILLY" (4) = "KARLIN" (5) = "BEV"	INITDB
QAM%	Selected mode for DE	1 = auto 0 = manual -1 = none	AUTO, CANCEL, INITDB, OPER
QFCN%	Selected function	0 = no key in effect 1 = START/ENABLE 2 = STOP/DISABLE 3 = DISPLAY DIAGRAM 4 = SET POINT/LIMITS 5 = AUTO 6 = PRINT REPORT 7 = MODIFY SCHED 8 = CHANGE OPER 9 = CONFIRM ACTION 10 = CANCEL ACTION	INITDB, FKEYIN, CONFIRM, XDD,OPER, CANCEL, DSCHED, GONOGO, DIAGRAM, SETPT, AUTO, REPORT, SCHED
QMENU%	Selected menu item	$1 \leq$ QMENU% $\leq$ NBOXES%	INITDB, DIAGRAM, REPORT, MSIN, MOIN, MLIN, MRIN, REPOUT MDIN, OPER, CONFIRM, SCHED, CANCEL, SETPT, XDD, MENUIN, DMENU, XMS, XPR

TABLE 5
(continued)

MMI DATA BASE

Variable	Function	Values	Set By
QOPER%	Current operator	1-NOPER%	OPER, XCHOP
QPT%	Selected point	1-NRTPTS% = point	INITDB, GONOGO, SETPT, OPER, XSPL, XSE,XSD, SYMBIN, CANCEL, CONFIRM
QSCH%	Selected Schedule	0 = None 1-NDES% = schedule	SCHED CANCEL, XMS,MSIN, INITDB
RODS%	Report output device selection variable	0 = none 1 = printer 2 = CRT	REPOUT INITDB, REPORT,XPR
S%	Numerical Value of Seconds	Current Seconds past the minute. 0-59	GJTS, TIME
S9%	Temporary seconds storage	Same as S%	IJTS
SALARM%	Flag to determine whether or not to sound the alarm tone	0 = do not sound tone 1 = sound tone	INITDB, ADISABLE, ANNUNC
SCHTMP$(25)	Temporary storage for changing schedule	Schedule values that are currently being modified.	DSCHED MSCHED
SHUTDOWN%	Flag to stop system (shutdown)	1 = shutdown 0 = do not shutdown	INITDB, XCHOP
SID$(23)	String array which contains the symbol identifier.	ADPT%/SSED% (0) = FID 1/0 (1) = MUX 1 1/0 (2) = MUX 2 1/0	INITDB

TABLE 5
(continued)

MMI DATA BASE

Variable	Function	Values	Set By
SID$(23) (continued)	NOTES: 1. SID$(0) to SID$(4) are control points. 2. SID$(5) to SID$(14) are monitor (alarm) points. 3. SID$ (15) to SID (23) are indicator points. 4. ADPT% => analog (0) or digital (1). 5. SSED% => start/stop device (1) or enable/ disable device (0).	ADPT%/SSED% (3) = R 1/0 (4) = S 1/1 (5) = A 1/0 (6) = E 1/0 (7) = F 1/0 (8) = FL 1/0 (9) = LV 1/0 (10) = M 1/0 (11) = O 1/0 (12) = PS 0/0 (13) = T 0/0 (14) = W 0/0 (15) = DP 1/0 (16) = STR 1/1 (17) = TC 0/0 (18) = HW BOILER 1/1 (19) = PUMP 1/1 (20) = CHILLER 1/1 (21) = COOLING TOWER 1/1 (22) = AC 1/1 (23) = COIL 1/1	
SIM%	Number of the current simulation array. Simulation time period.	1-NNSIM%	INITDB UPDATE
SL% (23)	Array which contains the length (# of characters) of symbol string. NOTES: 1. This array is related to SID$ array. 2. SYMBOLS 0-2 are in a 30x30 pixel box. 3. Symbols 3-14 are in a 22 x 22 pixel box. 4. Symbols 15 and 16 are in a 28 x 18 pixel rectangle. 5. Symbol 17 is an 11 pixel radius circle. 6. Symbol 18 is a 46 x 60 pixel rectangle. 7. Symbol 19 is on 11 pixel circle with a 6 x 15 pixel rectangle.	(0) = 3 (FID) (1) = 5 (MUX 1) (2) = 5 (MUX 2) (3) = 1 (R) (4) = 1 (S) (5) = 1 (A) (6) = 1 (E) (7) = 1 (F) (8) = 2 (FL) (9) = 2 (LV) (10) = 1 (M) (11) = 1 (O) (12) = 2 (PS) (13) = 1 (T) (14) = 1 (W) (15) = 2 (DP) (16) = 3 (STR) (17) = 2 (TC) (18) = 9 (HW BOILER) (19) = 0 (20) = 7 (CHILLER)	INITDB

TABLE 5
(continued)

MMI DATA BASE

Variable	Function	Values	Set By
SL% (23) (continued)	8. Symbol 20 is a 76 x 64 pixel rectangle. 9. Symbol 21 has a 30 pixel base, 42 pixel top, and is 38 pixels high with a top rectangle 26 x 2 and inset 8 pixels, the bottom rectangle is 14 x 8 pixels and is inset 8 pixels and a 3 pixel semi-circle, on top. 10. Symbol 22 is a 28 x 28 pixel box. 11. Symbol 23 is a 8 x 20 pixel rectangle.	(21) = 0 (22) = 0 (23) = 0	
STMP	Temp. storage for set point		DMENU
TC#	Time control variable for synchronization of alarm tones at 5 second intervals.	Julian time at which bell should sound again.	INITDB, BUZOFF, MMIEXEC
TMP$	Temporary string storage		ANY ROUTINE
TPT%	Temporary point		ANY ROUTINE
TQUE%	Top of queue pointer		MMIEXEC, INITDB, ADISABLE
TT$(7,3)	String matrix used to store pertinent text for menus and schedules	(1,0) = "MENU OF DATA ENVIRONMENTS" (1,1) = "BUILDING 106 LOWER" (1,2) = "MAIN CHILLER PLANT" (1,3) = "SUPPLY NO. 2" (2,0) = "MENU OF ANALOG CONTROL SET POINT/LIMITS" (2,1) = "LOW OPERATING LIMIT" (2,2) = "SET POINT" (2,3) = "HIGH OPERATING LIMIT" (3,0) = "MENU OF ANALOG MONITOR SET POINT/LIMITS" (3,1) = "LOW ALARM LIMIT"	INITDB

TABLE 5 (continued)

MMI DATA BASE

Variable	Function	Values	Set By
TT$(7,3) (continued)		(3,2) = "SET POINT" (3,3) = "HIGH ALARM LIMIT" (4,0) = "MENU OF REPORTS" (4,1) = "ENERGY UTILIZATION SUMMARY" (4,2) = "LOCK OUT SUMMARY" (4,3) = "ALARM SUMMARY" (5,0) = "MENU OF OPERATIONS" (5,1) = "CHANGE OPERATOR" (5,2) = "HELP" (5,3) = "SYSTEM SHUTDOWN" (6,0) = "MENU OF SUPPORT DEVICES" (6,1) = "LINEPRINTER" (6,2) = "CRT SCREEN" (7,0) = "MENU OF SCHEDULES" (7,1) = "BUILDING 106 LOWER-Optimum Start/Stop Schedule" (7,2) = "MAIN CHILLER PLANT-Optimum Start/Stop Schedule" (7,3) = "SUPPLY NO. 2 - Optimum Start/Stop Schedule"	
UT%	Time to wait (in seconds) between real time updates.	Currently = 100	INITDB
VPS% (1,25)	Numerical matrix containing X and Y screen coordinates of Ith Schedule entry	(0,I) = X coordinate (1,I) = Y coordinate (0, 1) = 24 (0, 2) = 141 (0, 3) = 237 (0, 4) = 339 (0, 5) = 450 (0, 6) = 20 (0, 7) = 84 (0, 8) = 148 (0, 9) = 212 (0,10) = 276 (0,11) = 340 (0,12) = 404 (0,13) = 468 (0,14) = 20 (0,15) = 84 (0,16) = 148 (0,17) = 212 (0,18) = 276 (0,19) = 340 (0,20) = 404 (0,21) = 468	INITDB

TABLE 5
(concluded)

MMI DATA BASE

Variable	Function	Values	Set By
VPS% (1,25) (continued)		(0,22) = 77 (0,23) = 220 (0,24) = 271 (0,25) = 408 (1, 1) = 338 (1, 2) = 338 (1, 3) = 338 (1, 4) = 338 (1, 5) = 338 (1, 6) = 268 (1, 7) = 268 (1, 8) = 268 (1, 9) = 268 (1,10) = 268 (1,11) = 268 (1,12) = 268 (1,13) = 268 (1,14) = 238 (1,15) = 238 (1,16) = 238 (1,17) = 238 (1,18) = 238 (1,19) = 238 (1,20) = 238 (1,21) = 238 (1,22) = 150 (1,23) = 150 (1,24) = 150 (1,25) = 150	
XTUCH%	X-coordinate of touch panel input	0-511	TOUCHIN
Y%	Numerical value of year	Current Year 0-99	GJTS, TIME
Y9%	Temporary year storage	Same as Y%	IJTS
YTUCH%	Y-coordinate of touch panel input	0-511	TOUCHIN
ZIN$	Temporary string variable	Current keyboard Input	KBISR, XCHOP, MSCHED, MLIN, XPR

TABLE 6

REAL-TIME DATA BASE

Variable	Function	Values	Set By
ADPT% (NRTPTS%)*	"Analog" vs. "digital" point discriminator, where "analog" refers to a point which can assume a range of values, and "digital" refers to a point which can assume two discrete values representing opposite states	0 = analog 1 = digital	INITDB
ALARMVL (NRTPTS%)*	Alarm value for digital point I%		INITDB
ANPT%	Annunciation point (sent to ANNUNC)	1-NRTPTS%	UPDATE
APT%	The current alarm point, as defined by EMCS (never more than one at a time)	0 = none 1-NRTPTS% = point	INITDB UPDATE ANNUNC
ASTAT% (NRTPTS%)*	Alarm status flag for real-time data point I%	0 = point not in alarm 1 = point in alarm	UPDATE
CDBVL (NRTPTS%)*	Current (previous) value of real-time data point I%	> 0.0 (set by INITDB to -9999.0 to signify invalid data) if analog = current measurement if digital 0 = state value (off,low,etc.) 1 = state value (on,high,etc.)	INITDB UPDATE
CJT#	Current Julian Time		TIME GJTS
DSTAT% (NRTPTS%)*	Disable status flag for real-time data point I% indicates whether point should be processed or ignored	0 = point should be processed (is enabled) 1 = point is completely disabled (ignore it)	INITDB UPDATE
FDEAM%	Auto/manual flag for current data environment	1 = AUTO MODE (default) 0 = MANUAL MODE	INITDB, AUTO

TABLE 6
(continued)

REAL-TIME DATA BASE

Variable	Function	Values	Set By
HILIM (NRTPTS%)*	Highest non-alarm value for analog point I%	> LOWLIM (I%)	INITDB
IJT#	Incremented Julian Time		IJTS
LOWLIM (NRTPTS%)* (DE)	Lowest non-alarm value for analog point I%	$\geq$ 0.0	INITDB
LPNT% (NRTPTS%)*	Number of process diagram on which Ith point is located	$\geq$ 1	INITDB
NDES%	Number of DE's	1-limits of storage (currently 3)	INITDB
NRTPTS%	Number of real-time data points (note: for arrays listed below, I% assumes values of 1 to NRTPTS%)	1-limits of storage (currently 93)	INITDB
NWVL (NRTPTS%)*	Newest value of real-time data point I%	Same as the current simulation array (SIM**[NRTPTS%])	UPDATE
PNAME$ (0-23,0-2)	String matrix containing units, state values, and name of Ith point for text and message purposes.	(1,0) = digital point:0 state value or analog units (i,1) = digital point:1 state value (i,2) = point name (0,0) = "DISABLED" (0,1) = "ENABLED" (0,2) = "FID" (1,0) = "DISABLED" (1,1) = "ENABLED" (1,2) = "MUX #1" (2,0) = "DISABLED" (2,1) = "ENABLED" (2,2) = "MUX #2" (3,1) = DEGREES F." (3,1) = "" (3,2) = "Temperature Controller Reset Interface" (4,0) = "STOPPED" (4,1) = "STARTED"	INITDB

TABLE 6
(continued)

REAL-TIME DATA BASE

Variable	Function	Values	Set By
PNAME$ (continued)		(4,2) = "Start/Stop Control Interface"	
		(5,0) = "NO ALARM"	
		(5,1) = "ALARM DETECTED"	
		(5,2) = "Alarm Contact Signal Monitor"	
		(6,0) = "???"	
		(6,1) = ""	
		(6,2) = "Enthalpy/Economizer Control Interface"	
		(7,0) = "GPM"	
		(7,1) = ""	
		(7,2) = "Flow Indication Monitor"	
		(8,0) = "NO FLAME"	
		(8,1) = "FLAME PRESENT"	
		(8,2) = "Flame Indication Monitor"	
		(9,0) = "LOW"	
		(9,1) = "HIGH"	
		(9,2) = "Level Indication Monitor"	
		(10,0)= "KW"	
		(10,1)= ""	
		(10,2)= "Meter"	
		(11,0)= "OFF/NONE/OPEN"	
		(11,1)= "ON/POSITIVE/CLOSED"	
		(11,2)= "On/Off Status Monitor"	
		(12,0)= "PERCENT OPEN"	
		(12,1)= ""	
		(12,2)= "Position Sensor"	
		(13,0)= "Degrees F."	
		(13,1)= ""	
		(13,2)= "Temperature Monitor"	
		(14,0)= "%RH"	
		(14,1)= ""	
		(14,2)= "Humidity Monitor"	
		(15,0)= "OPEN"	
		(15,1)= "CLOSED"	
		(15,2)= "Differential Pressure Switch"	
		(16,0)= "STOPPED"	
		(16,1)= "STARTED"	
		(16,2)= "Motor Starter"	
		(17,0)= "DEGREES F."	
		(17,1)= ""	
		(17,2)= "Temperature Controller"	
		(18,0)= "STOPPED"	
		(18,1)= "STARTED"	

TABLE 6
(continued)

REAL-TIME DATA BASE

Variable	Function	Values	Set By
PNAME$ (concluded)		(18,2)= "Hot Water Boiler" (19,0)= "STOPPED" (19,1)= "STARTED" (19,2)= "Pump" (20,0)= "STOPPED" (20,1)= "STARTED" (20,2)= "Chiller" (21,0)= "STOPPED" (21,1)= "STARTED" (21,2)= "Cooling Tower" (22,0)= "STOPPED" (22,1)= "STARTED" (22,2)= "Unit Air Conditioner" (23,0)= "NOT IN USE" (23,1)= "IN USE" (23,2)= "Coil"	
PHIER% (NRTPTS%)*	Next higher point to which Ith point is connected in the process hierarchy	0 = is highest point N = attached to Nth point	INITDB
PTYPE% (NRTPTS%)*	Ith point type indicator (symbol type)	refers to type identified in SID$ (It is a negative value if it has any points lower than it in the heirarchy.)	INITDB
PX% (NRTPTS%)*	X-coordinate (left) of location of Ith point's symbol on process diagram	0-511	INITDB
PY% (NRTPTS%)*	Y-coordinate (lower) of location of Ith point's symbol on process diagram	0-511	INITDB
S$ (NDES%,NSE%)	Schedules for each DE are stored here NOTE: S$(1,1-25) is the same as S$(2-3,1-25).	(1, 1) = "68.2F" (1, 2) = "0730" (1, 3) = "78.1F" (1, 4) = "89.4F" (1, 5) = "CLG" (1, 6) = "NA" (1, 7) = "0800" (1, 8) = "0800" (1, 9) = "0800"	INITDB XMS

TABLE 6
(continued)

REAL-TIME DATA BASE

Variable	Function	Values	Set By
S$ (continued)		(1,10) = "0800"	
		(1,11) = "0800"	
		(1,12) = "NA"	
		(1,13) = "NA"	
		(1,14) = "NA"	
		(1,15) = "1700"	
		(1,16) = "1700"	
		(1,17) = "1700"	
		(1,18) = "1700"	
		(1,19) = "1700"	
		(1,20) = "NA"	
		(1,21) = "NA"	
		(1,22) = "S"	
		(1,23) = "65F"	
		(1,24) = "80F"	
		(1,25) = "5.00"	
		(2, 1) = "68.2F"	
		(2, 2) = "0730"	
		(2, 3) = "78.1F"	
		(2, 4) = "89.4F"	
		(2, 5) = "CLG"	
		(2, 6) = "NA"	
		(2, 7) = "0800"	
		(2, 8) = "0800"	
		(2, 9) = "0800"	
		(2,10) = "0800"	
		(2,11) = "0800"	
		(2,12) = "NA"	
		(2,13) = "NA"	
		(2,14) = "NA"	
		(2,15) = "1700"	
		(2,16) = "1700"	
		(2,17) = "1700"	
		(2,18) = "1700"	
		(2,19) = "1700"	
		(2,20) = "NA"	
		(2,21) = "NA"	
		(2,22) = "S"	
		(2,23) = "65F"	
		(2,24) = "80F"	
		(2,25) = "5.00"	
		(3, 1) = "68.2F"	
		(3, 2) = "0730"	
		(3, 3) = "78.1F"	
		(3, 4) = "89.4F"	
		(3, 5) = "CLG"	

TABLE 6
(concluded)

REAL-TIME DATA BASE

Variable	Function	Values	Set By
S$ (concluded)		(3, 6) = "NA" (3, 7) = "0800" (3, 8) = "0800" (3, 9) = "0800" (3,10) = "0800" (3,11) = "0800" (3,12) = "0800" (3,13) = "NA" (3,14) = "NA" (3,15) = "1700" (3,16) = "1700" (3,17) = "1700" (3,18) = "1700" (3,19) = "1700" (3,20) = "NA" (3,21) = "NA" (3,22) = "S" (3,23) = "65F" (3,24) = "80F" (3,25) = "5.00"	
SCHTMP$ (NSE%)	Temporary storage for schedule being modified		DSCHED
SIMI (NRTPTS%)* -SIM28 (NRTPTS)*	SIMULATION arrays	Contain the current simulated readings for all real time data points in system.	UPDATE
SPT(NRTPTS%)*	Set point for analog point I%	$\leq$ HILIM (I%) $\geq$ LOWLIM (I%)	INITDB, XSPL
SSED% (NRTPTS%)*	Indicate device status. Start/Stop - Enable/Disable device discriminator array.	0 = Enable/disable device 1 = Start/stop device	INITDB
UC#	Real time update, synchronization variable	(Year-Julian Day-HOURS-MINUTES-SECONDS)	TIME

* See Appendix F for a listing of all real-time database arrays.

3.2 Module Descriptions

This section includes written descriptions and flowcharts for program modules contained in the MMI software system. The descriptions are arranged in hierarchial order, corresponding to the representation in Figure 12.

NAME: MMI EXEC

PURPOSE:

MMI EXEC serves as the executive program for the system. It is used to supervise and coordinate the receipt and processing of command inputs and real-time data.

OPERATIONAL DESCRIPTION:

MMI EXEC consists of a series of calls to appropriate subroutines which process operator commands and real-time data updates. MMI EXEC also drives the real-time EMCS data update simulator. The simulator emulates operation of an interrupt-driven system.

When MMI EXEC is initiated, it causes the MMI data base to be initialized, runs the HELP display sequence, and gets the simulated initial real-time data base values. Once the system has been initialized, MMI EXEC waits for an interrupt. An interrupt can come from the touch panel, the keyboard, or the EMCS computer simulation. Once an interrupt has been detected, MMI EXEC calls the appropriate interrupt service routine to handle that interrupt. MMI EXEC also performs the "housekeeping chores" of displaying the current time and sounding the alarm tone when appropriate.

CALLED BY:

The program is initiated by the user.

NAME: MMI EXEC (continued)

CALLS:

DISMESS

GJTS

HELP

IJTS

INITDB

TIME

TOUCHIN

TPDRIVER

UPDATE

PASSED ARGUMENTS:

(none)

RETURNED ARGUMENTS:

DSTAT%	- Disable Status
MDEX%	- Message index (to DISMESS)
TQUE%	- Top of Alarm Queue Pointer
ZIN$	- Keyboard input response

FILE INPUT/OUTPUT:

(none)

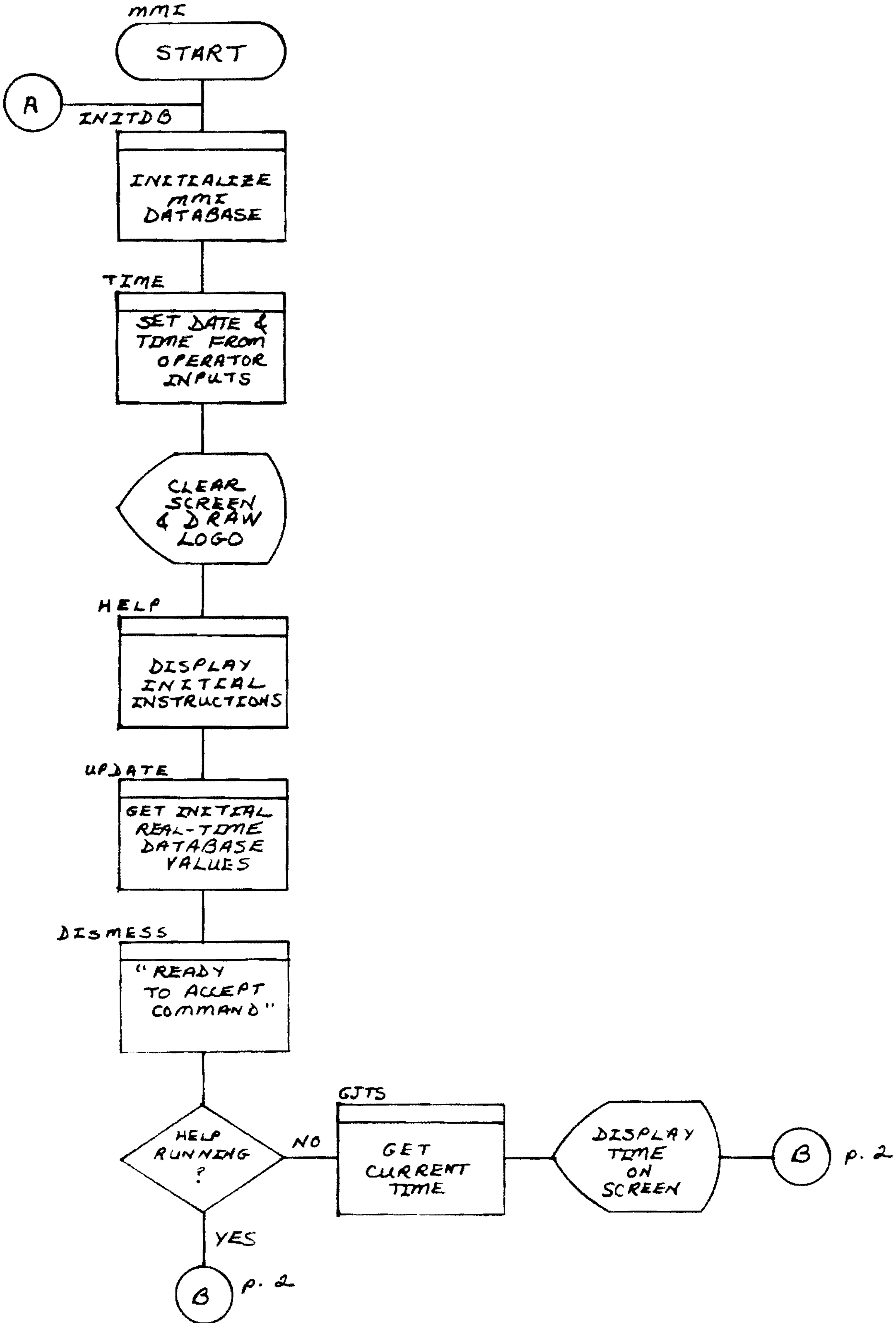
HARDWARE INTERACTION:

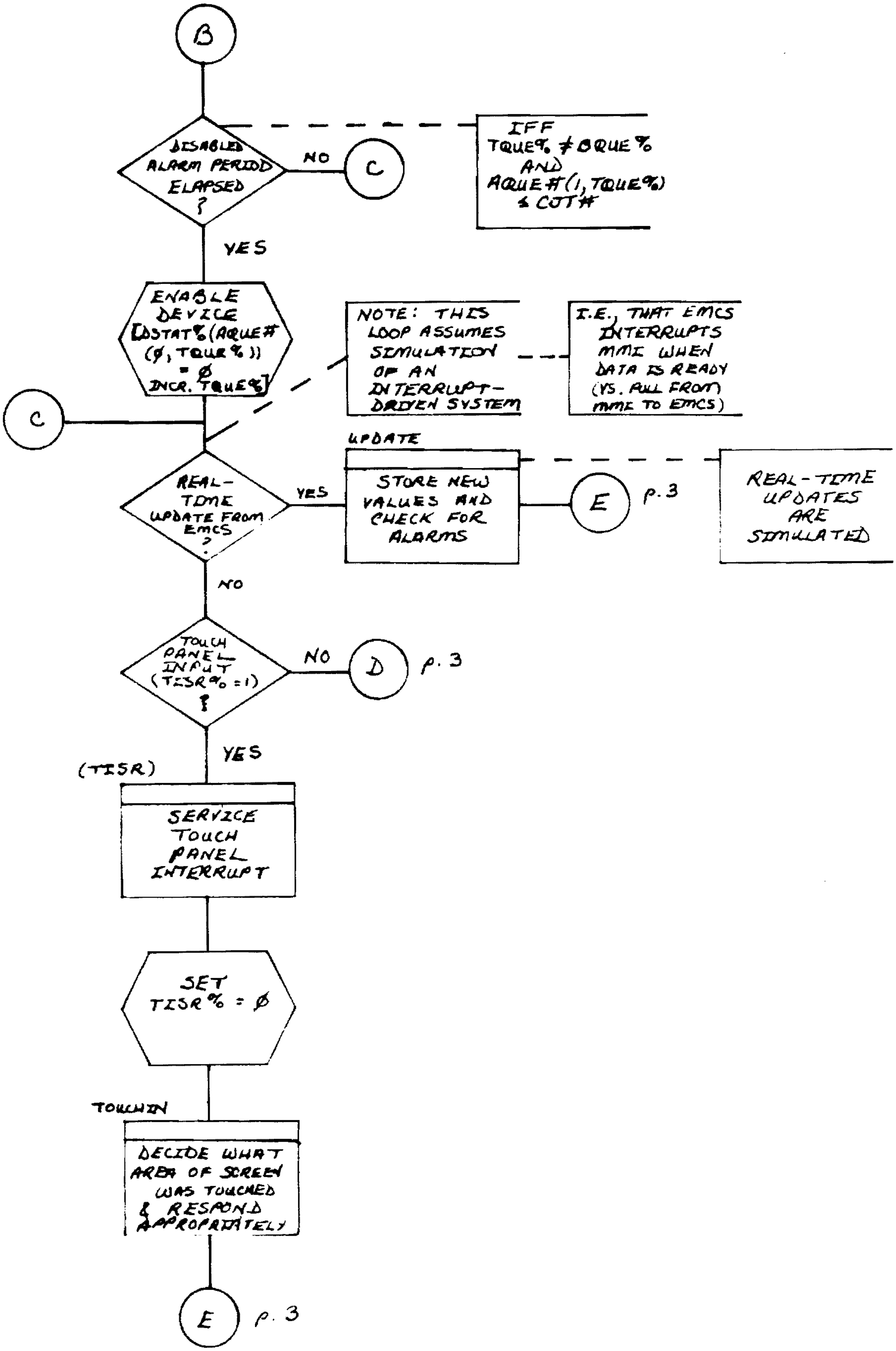
Touch panel	- determine if a "hit" has occurred
Keyboard	- determine if keyboard input is attempted Memorysemaphores (i.e., flags set by interrupt service routines)

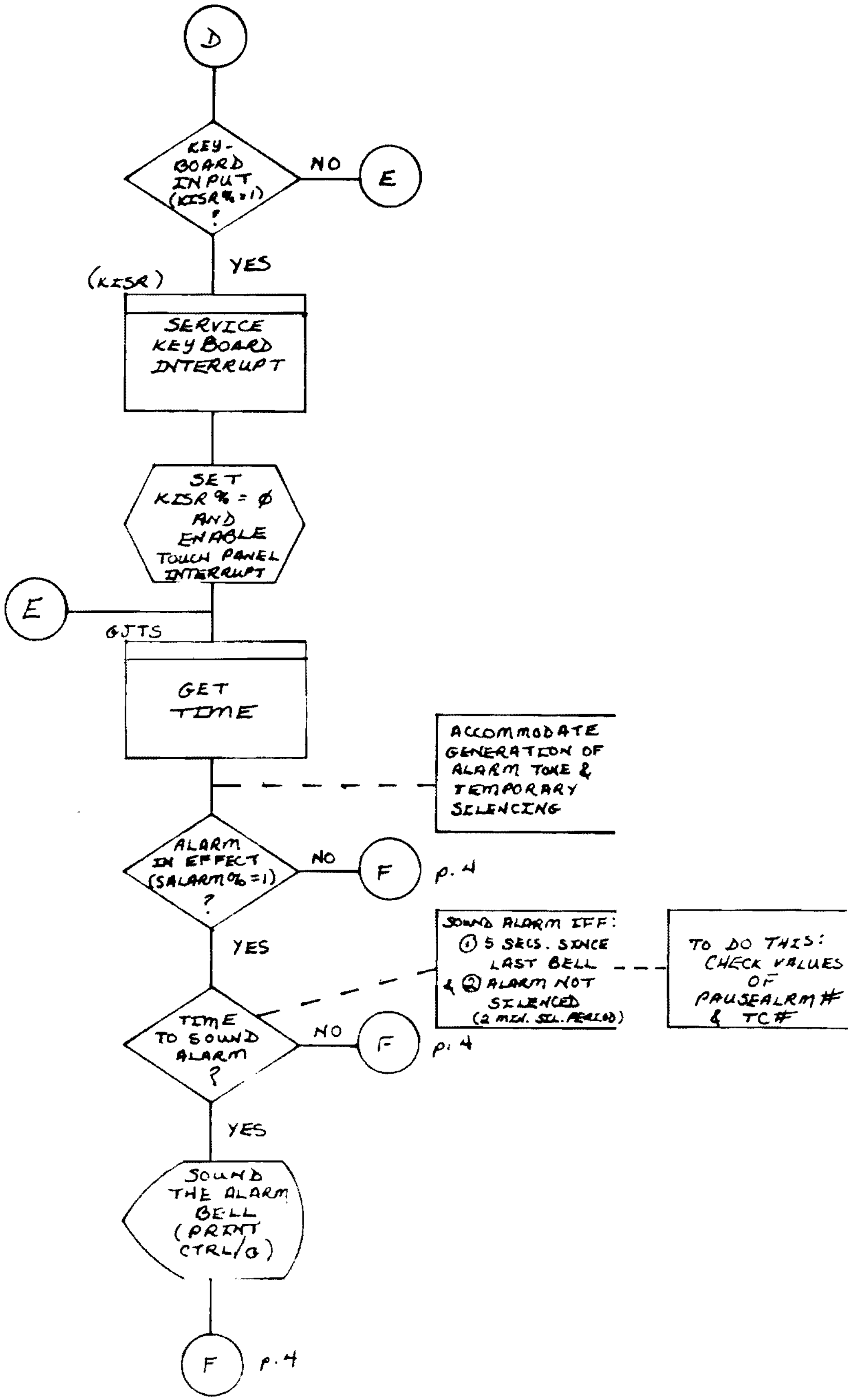
NAME: MMI Exec (concluded)

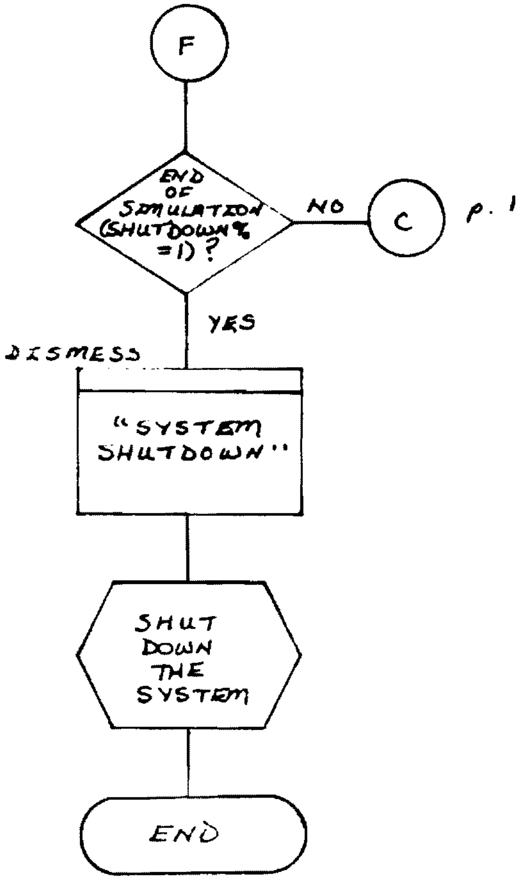
DESIGN NOTES:

The real-time data simulator provides the MMI with real-time data base values in the absence of a link to the EMCS computer. The simulator uses a countdown timer to initiate a call to the UPDATE module where a set of new database values are read in from disk storage. Thus, the simulation consists of a set of "table-driven" devices.


MMI
START
A
INITDB
INITIALIZE MMI DATABASE
TIME
SET DATE & TIME FROM OPERATOR INPUTS
CLEAR SCREEN & DRAW LOGO
HELP
DISPLAY INITIAL INSTRUCTIONS
UPDATE
GET INITIAL REAL-TIME DATABASE VALUES
DISMESS
"READY TO ACCEPT COMMAND"
HELP RUNNING ?
NO
GJTS
GET CURRENT TIME
DISPLAY TIME ON SCREEN
B
p. 2
YES
B
p. 2


B
DISABLED ALARM PERIOD ELAPSED ?
NO
C
IFF TQUE% ≠ BQUE % AND AQUE#(1,TQUE%) ≤ CJT#
YES
ENABLE DEVICE [DSTAT%(AQUE#(Ø,TQUE%)) = Ø INCR. TQUE%]
NOTE: THIS LOOP ASSUMES SIMULATION OF AN INTERRUPT-DRIVEN SYSTEM
I.E., THAT EMCS INTERRUPTS MMI WHEN DATA IS READY (VS. POLL FROM MMI TO EMCS)
C
UPDATE
REAL-TIME UPDATE FROM EMCS ?
YES
STORE NEW VALUES AND CHECK FOR ALARMS
E
p. 3
REAL-TIME UPDATES ARE SIMULATED
NO
TOUCH PANEL INPUT (TISR% = 1) ?
NO
D
p. 3
YES
(TISR)
SERVICE TOUCH PANEL INTERRUPT
SET TISR% = Ø
TOUCHIN
DECIDE WHAT AREA OF SCREEN WAS TOUCHED & RESPOND APPROPRIATELY
E
p. 3


D
KEY-BOARD INPUT (KISR%=1) ?
NO
E
YES
(KISR)
SERVICE KEYBOARD INTERRUPT
SET KISR% = Ø AND ENABLE TOUCH PANEL INTERRUPT
E
GJTS
GET TIME
ACCOMMODATE GENERATION OF ALARM TONE & TEMPORARY SILENCING
ALARM IN EFFECT (SALARM%=1) ?
NO
F
p. 4
YES
SOUND ALARM IFF: ① 5 SECS. SINCE LAST BELL & ② ALARM NOT SILENCED (2 MIN. SIL. PERIOD)
TO DO THIS: CHECK VALUES OF PAUSEALRM# & TC#
TIME TO SOUND ALARM ?
NO
F
p. 4
YES
SOUND THE ALARM BELL (PRINT CTRL/G)
F
p. 4


F
END OF SIMULATION (SHUTDOWN% =1)?
NO
C
p. 1
YES
DISMESS
"SYSTEM SHUTDOWN"
SHUT DOWN THE SYSTEM
END

NAME: INITDB

PURPOSE:

The INITDB subroutine is used to set the initial values of all variables in the MMI data base.

OPERATIONAL DESCRIPTION:

There are two major data bases in the system: the MMI data base contains variables which pertain to software system operation, whereas the real-time data base contains current values which pertain to the (simulated) real-time system. The INITDB subroutine explicitly sets the value of each variable in the man-machine interface data base. INITDB then sets constants (flags, counters, names, etc) for the real-time data base. INITDB also sets the real-time data base ready flag, DBRDY%, to 0 indicate that the real-time data base has not yet been initialized. INITDB then returns to MMI EXEC.

CALLED BY:

MMI EXEC

CALLS:

(none)

PASSED ARGUMENTS:

(none)

NAME: INITDB (continued)

RETURNED ARGUMENTS:

ADPT%	- Analog/Digital Point discriminator array
ALARMVL	- Alarm Value for digital point array
ANPT%	- Annunciation Point
APT%	- Current Alarm Point
AQUE#	- Alarm disablement queue
ASTAT%	- Alarm status flag for EMCS point array
BOXLOC%	- Menu box coordinate
BQUE%	- Bottom of queue pointer plus 1
CAL% (1-8)	- Touch panel calibration constants
CDBVL	- Current (previous) value of EMCS point array
CJT#	- Current Julian Time
D%	- Numerical Value of Date
D9%	- Temporary date storage
DBRDY%	- Real time data base ready flag
DIAMEN%	- Currently displayed diagram or menu
DSTAT%	- Disable status flag for EMCS Point array
FDEAM%	- Auto/Manual flag for current DE
FKLOC%	- Function key coordinates
GDTYPE%	- Graphics display type
H%	- Numerical value of hour
H9%	- Temporary storage for numerical value of hour
HC%	- Time inhinit for help. Control variable
HILIM	- Highest non-alarm value for EMCS analog point array
IJT#	- Incremented Julian Time
JDAY%	- Julian Day matrix
KCOLR%	- Foreground/Background function key colors
KISR%	- Keyboard input flag
KTEXT$	- Function key text string matrix
LALARM%	- Location of alarm (monitor) point types in SID$ array
LCNT%	- Location of control point types in SID$ array
LDE%	- Number of last DE displayed

NAME: INITDB (continued)

RETURNED ARGUMENTS: (continued)

LIND%	- Location of indicator point types in SID$ array
LOWLIM	- Lowest non-alarm value for EMCS analog point array
LPNT%	- Process Diagram (DE) on which EMCS point is located array
M%	- Numerical value of minutes
M1%	- Leap year indicator flag
M2%	- Numerical value of the day of the month
M9%	- Temporary storage for numerical value o f minutes
MQUE%	- Maximum number of alarms in alarm queue
MT%	- Numerical value of the month
NALARM%	- Number of monitor (alarm) point types
NBOXES%	- Number of menu boxes for menus
NCNT%	- Number of control point types
NDES%	- Number of Data Environments (DE)
NHELPBF%	- Number of HELP buffer files
NIND%	- Number of indicator point types
NKEYS%	- Number of function keys
NMESS%	- Number of messages in DISMESS master file
NNSIM%	- Number of simulation arrays
NOPER%	- Number of system operators
NSE%	- Number of schedule entries
NRTPTS%	- Number of real-time EMCS points
NUMREPTS%	- Number of reports
NWVL	- Newest value of EMCS point array
OLEVEL%	- Security level of operators
OPER$	- String array containing operator's name
PAUSEALRM#	- Pause timer for alarm tone
PHIER%	- Array structured to show how points on DE are tied together
PNAME$	- String matrix containing text for each point type
PTYPE%	- Point type indicator array
PWRD$	- Password string array
PX%	- X-coordinate for EMCS point (symbol)
PY%	- Y-coordinate for EMCS point (symbol)

NAME: INITDB (continued)

RETURNED ARGUMENTS: (continued)

QAM%	- Selected mode for DE
QFCN%	- Selected function
QMENU%	- Selected menu item
QOPER%	- Current system operator
QPT%	- Selected point
QSCH%	- Selected schedule
RODS%	- Report Output Device Selection variable
S$	- Schedules for DE's
S%	- Numerical value of seconds
S9%	- Temporary storage for numerical value of seconds
SALARM%	- Flag to determine whether or not to sound alarm
SCHTMP$	- Temporary schedule storage
SHUTDOWN%	- Flag to stop system
SID$	- Symbol identifier string array
SIM%	- Current simulation time period
SL%	- Symbol string identifier length array
SPT	- Set point value for EMCS analog point array
SSED%	- Start/Stop-Enable/Disable device discriminator array
T%	- Keyboard input variable used in KB ISR
TC#	- Time control variable for synchronization of alarm tones
TPT%	- Temporary EMCS point number storage
TQUE%	- Top of alarm queue pointer
TT$	- Matrix containing pertinent text for DE's
UC#	- Real-time update synchronization variable
UT%	- Time to wait in seconds between real-time updates
Y%	- Numerical value of year
Y9%	- Temporary storage for numerical value of year
VPS%	- Schedule entry coordinates
ZC%	- Internal control variable used when modifying schedule
ZIN$	- Raw keyboard input string

(see flowchart for description of initial values)

NAME: INITDB (concluded)

FILE INPUT/OUTPUT:

Open DISMESS message file.

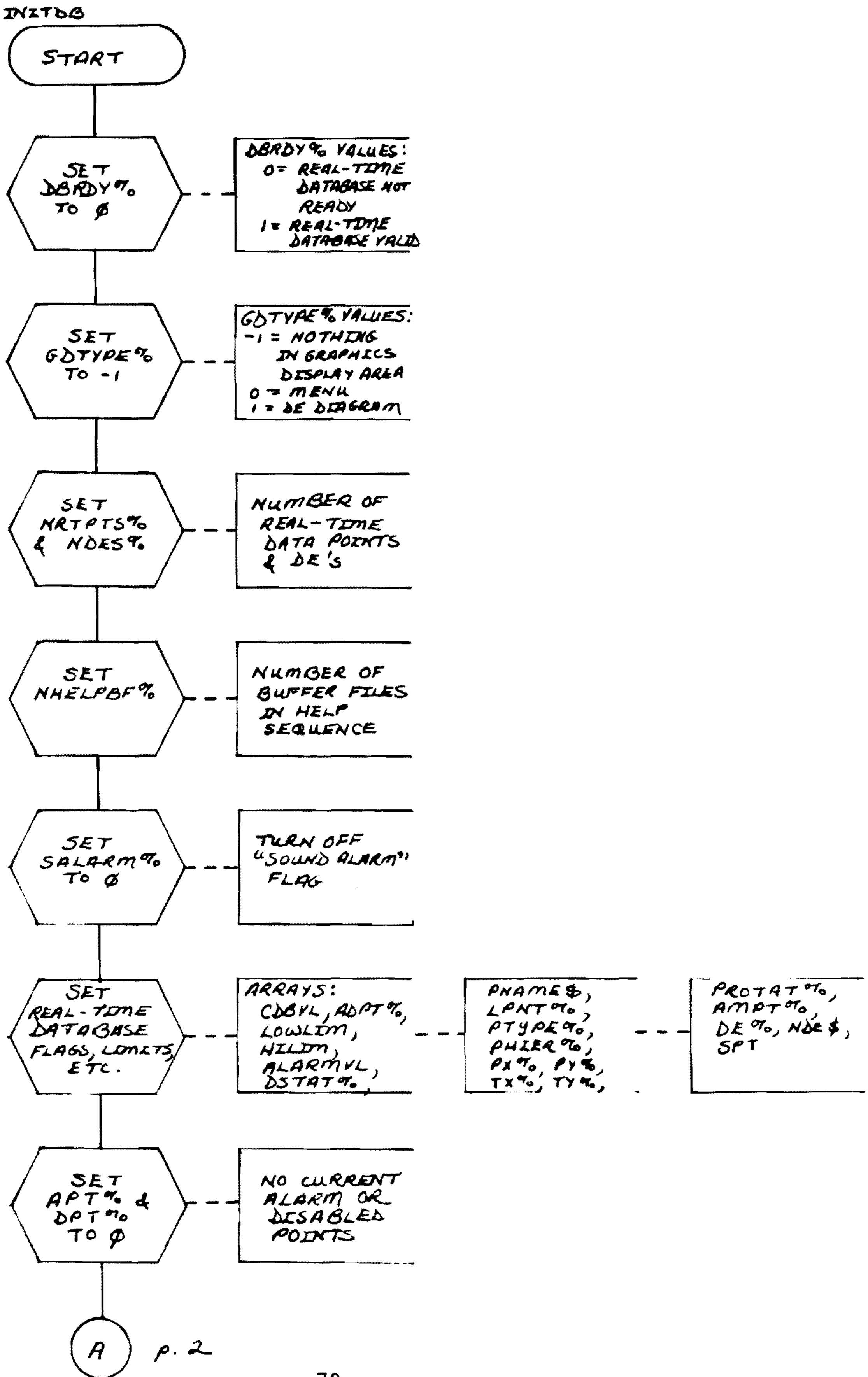
HARDWARE INTERACTION:

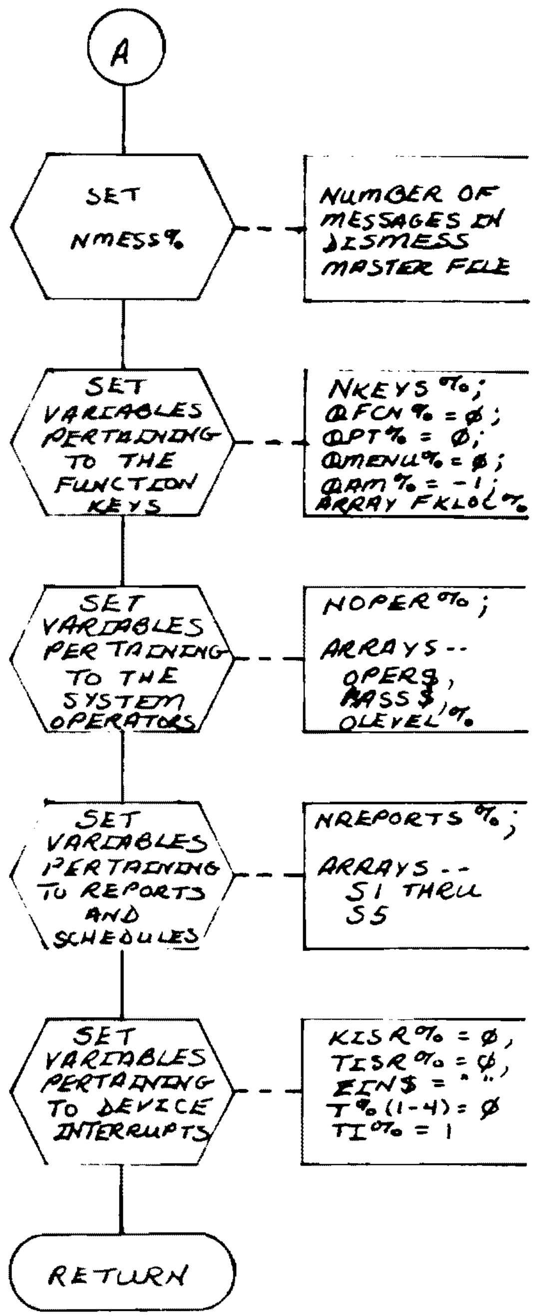
Disk Drive - used to read in initial values for some array structures.

DESIGN NOTES:

(none)

INITDB: INITIALIZE MMI DATABASE




A
SET NMESS%
NUMBER OF MESSAGES IN DISMESS MASTER FILE
SET VARIABLES PERTAINING TO THE FUNCTION KEYS
NKEYS%; @FCN% = Ø; @PT% = Ø; @MENU% = Ø; @AM% = -1; ARRAY FKLOC%
SET VARIABLES PERTAINING TO THE SYSTEM OPERATORS
NOPER%; ARRAYS -- OPER$, PASS$, OLEVEL%
SET VARIABLES PERTAINING TO REPORTS AND SCHEDULES
NREPORTS%; ARRAYS -- S1 THRU S5
SET VARIABLES PERTAINING TO DEVICE INTERRUPTS
KISR% = Ø, TISR% = Ø, EIN$ = "", T%(1-4) = Ø, TI% = 1
RETURN

NAME: HELP

PURPOSE:

The HELP subroutine is used to provide a series of screen displays to the operator which contain helpful information describing how to operate the system.

OPERATIONAL DESCRIPTION:

HELP first disables the touch panel and enables the keyboard for input. The variable NHELPBF% contains the number of HELP screen displays available. The subroutine enters a loop which is executed NHELPBF% times. Each time through the loop, a filename is constructed of the form "HLP" + (number of loop), (e.g., HLP1, HLP2, etc.). These filenames correspond to disk files containing the memory image of a HELP display. The file is then read into a memory buffer, from which it is displayed on the screen. The subroutine then waits until the operator presses the Return key before going on to the next display in the sequence. The sequence can be prematurely stopped by typing an S followed by the Return key from the keyboard. When the sequence is complete, the keyboard is disabled and program control is transfered back to the calling routine.

CALLED BY:

MMI Exec at the beginning of system operation.

XCHOP

CALLS:

Keyboard Interrupt Service Routine (KB ISR) in MMI EXEC

DISMESS

NAME: HELP (continued)

PASSED ARGUMENTS:

NHELPBF% - Number of HELP messages

ZIN$ - Keyboard input response

RETURNED ARGUMENTS:

GDTYPE% - Graphics display type indicator

HC% - Help control variable

FILE INPUT/OUTPUT:

The subroutine retrieves disk files of the form HLP**, where ** is an integer in the range 1 through NHELPBF%.

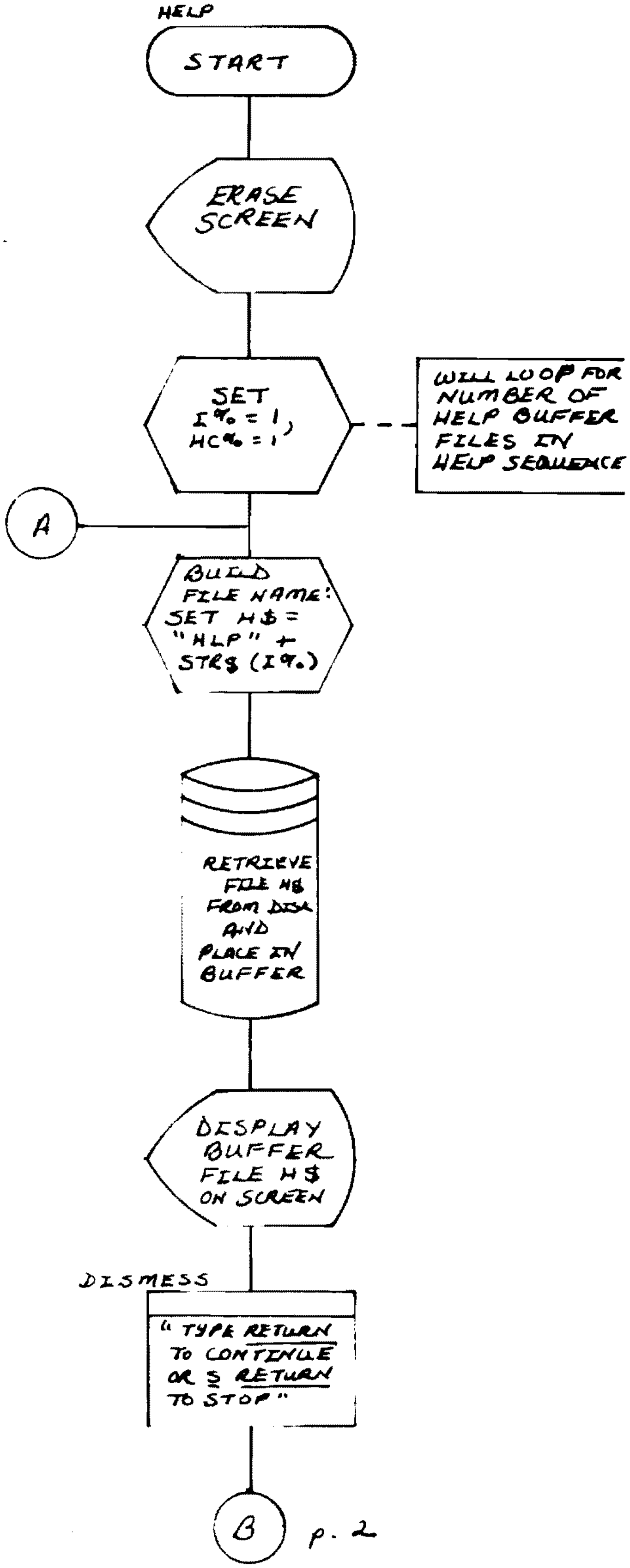
HARDWARE INTERACTION:

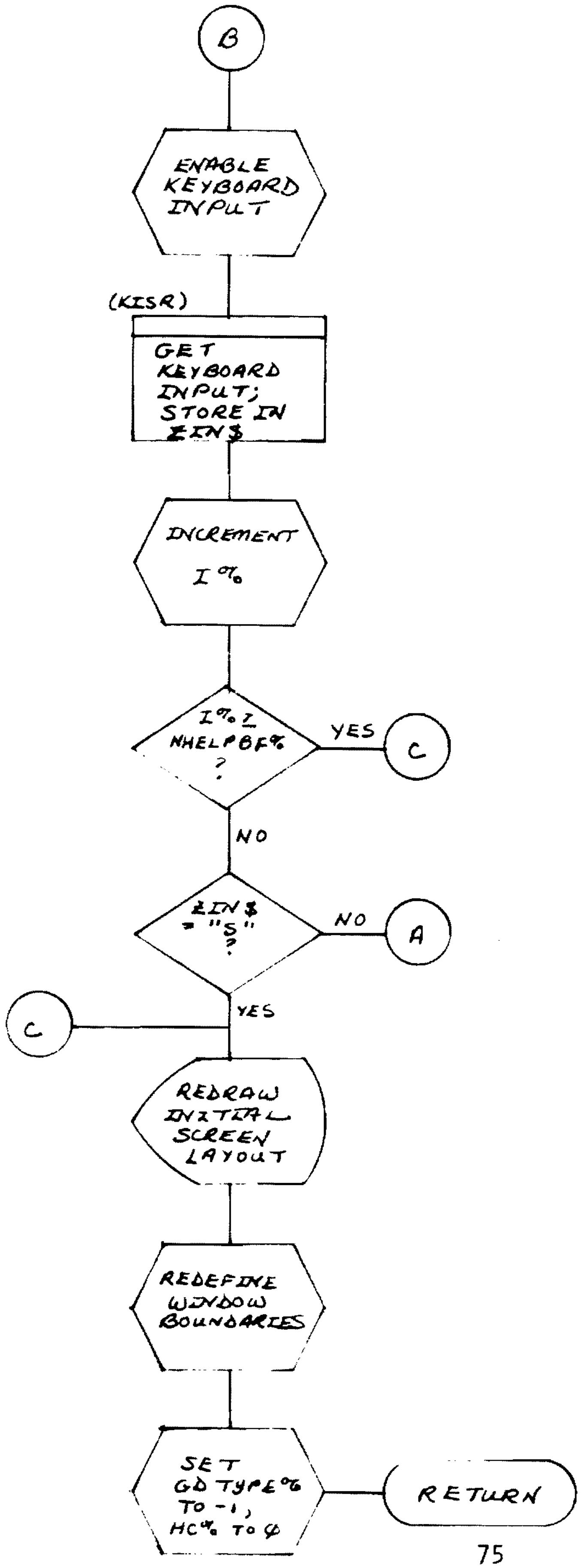
CRT DISPLAY UNIT - used to display information

KEYBOARD - used to input operator commands

DESIGN NOTES:

(none)


HELP
START
ERASE SCREEN
SET I% = 1, HC% = 1
WILL LOOP FOR NUMBER OF HELP BUFFER FILES IN HELP SEQUENCE
A
BUILD FILE NAME: SET H$ = "HLP" + STR$ (I%)
RETRIEVE FILE H$ FROM DISK AND PLACE IN BUFFER
DISPLAY BUFFER FILE H$ ON SCREEN
DISMESS
"TYPE RETURN TO CONTINUE OR S RETURN TO STOP"
B
p. 2


B
ENABLE KEYBOARD INPUT
(KISR)
GET KEYBOARD INPUT; STORE IN ZIN$
INCREMENT I%
I% ≥ NHELPBF% ?
YES
C
NO
ZIN$ = "S" ?
NO
A
YES
C
REDRAW INITIAL SCREEN LAYOUT
REDEFINE WINDOW BOUNDARIES
SET GDTYPE% TO -1, HC% TO φ
RETURN

NAME: UPDATE

PURPOSE:

The UPDATE subroutine is used to retrieve new real-time data base values as they are "updated" (i.e., generated by the simulator) and to check them for alarm conditions.

OPERATIONAL DESCRIPTION:

The subroutine gets new values for each point and stores them in an array. UPDATE first checks each new value to see if it is different from its previous value. If the values are different the new value replaces the old value and an alarm check is made. To perform the alarm check, the subroutine first determines whether or not the point is a monitor point. If it is not a monitor point then it is ignored. Otherwise, a check is made to determine whether the point is analog or digital and then a comparison is made between its new value and the alarm value or range specified for that point. If the point is found to be in alarm condition and the alarm has not already been sounded for that point, the ANNUNC subroutine is called to sound the alarm. If the point is not found to be in alarm condition, but was in alarm prior to the receipt of the new value, ANNUNC is called to silence the alarm for that point. If no change in alarm status has occurred, no annunciation is made. When all new values have been read and checked, the real-time data base ready flag (DBRDY%) is set and program control returns to the calling routine.

NAME: UPDATE (continued)

CALLED BY:

MMI EXEC

CALLS:

ANNUNC

PASSED ARGUMENTS:

ADPT%	- Analog versus digital point discriminator
ALARMVL%	- Alarm value for digital point
ASTAT%	- Alarm status flag for real-time data point
CDBVL	- Current (previous) value of real-time data point
DIAMEN%	- Currently displayed diagram or menu
DSTAT%	- Disable status flag for real-time data point
GDTYPE%	- Graphics Display Type
HILIM	- Highest non-alarm value for analog point
LALARM%	- Location of alarm (monitor) point types in SID$ array
LOWLIM	- Lowest non-alarm value for analog point
LPNT%	- Process Diagram (DE) on which EMCS point is located array
NALARM%	- Number of monitor (alarm) point types
NNSIM%	- Number of simulation arrays
NRTPTS%	- Number of real-time data points
PNAME$	- Textal name of point
PTYPE%	- Point type indicator array
SIM%	- Current simulation time period
SSED%	- Start/Stop - Enable/Disable device discriminator array
UT%	- Time to wait in seconds between real-time updates

NAME: UPDATE (continued)

RETURNED ARGUMENTS:

ANPT%	-	Annunication point
APT%	-	Current alarm point
ASTAT%	-	Alarm status flag for real-time point
CDBVL	-	Current value of real-time data point
SIM%	-	Current simulation time period
UC#	-	Real time update synchronization variable

FILE INPUT/OUTPUT:

Reads simulation values from disk array files of the form SIM**, where ** is an interger in the range 1 through NNSIM%.

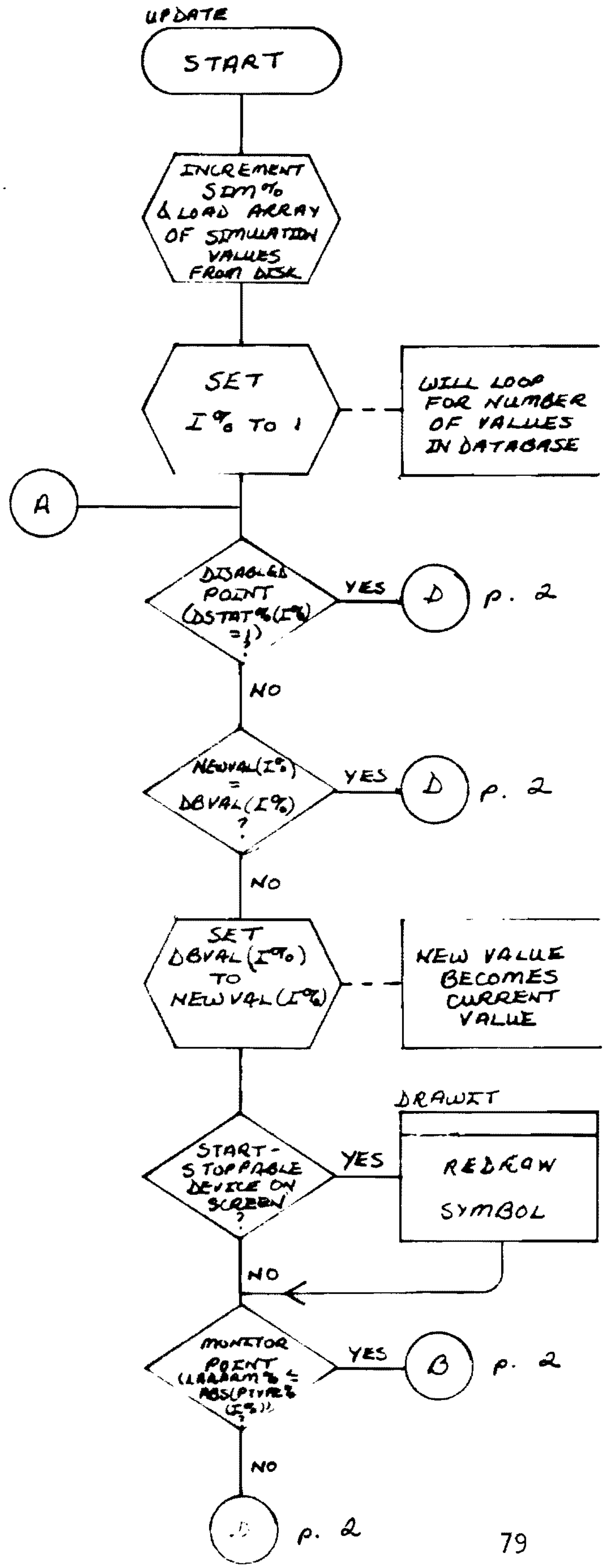
HARDWARE INTERACTION:

(none)

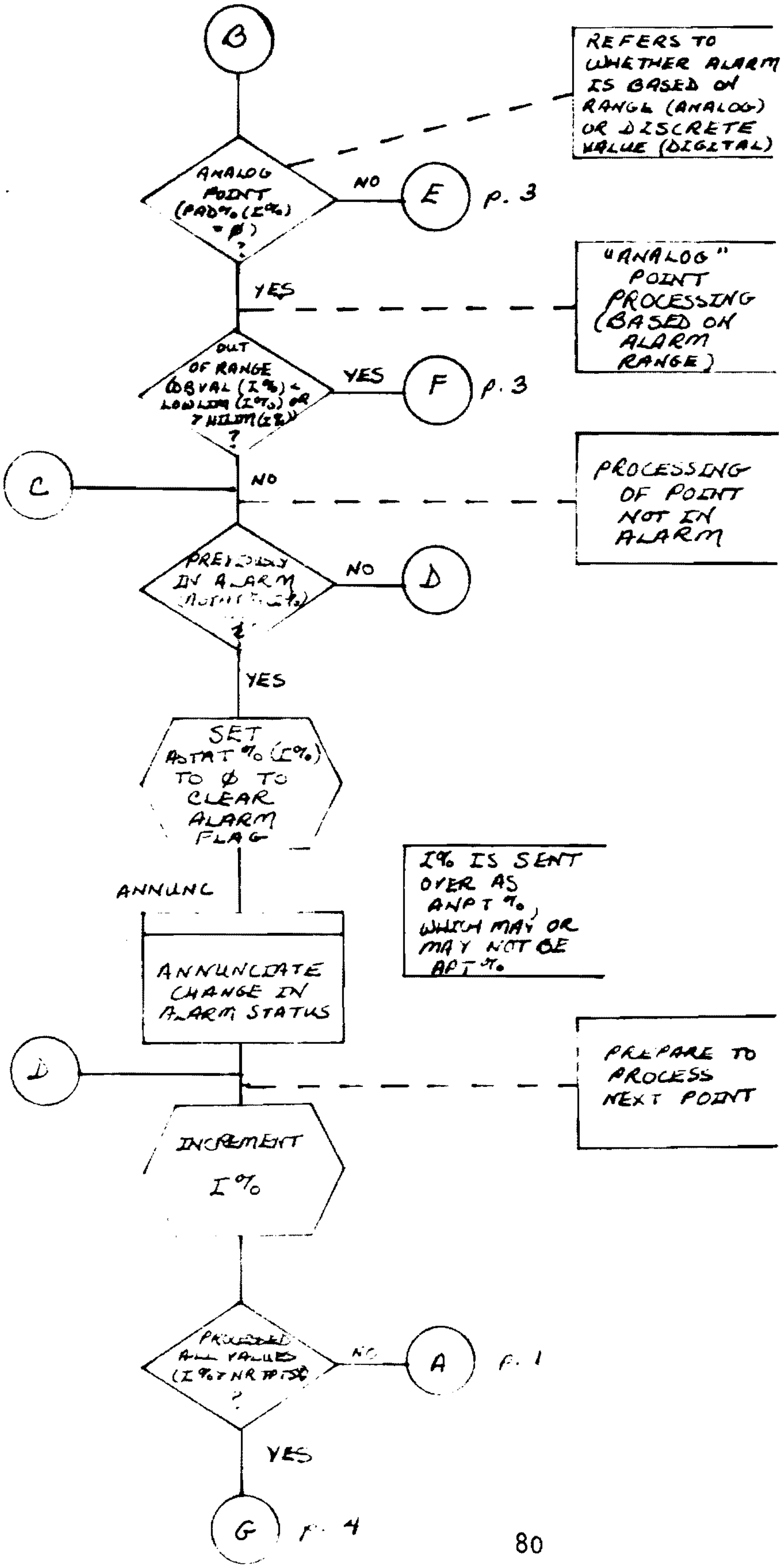
DESIGN NOTES:

(none)

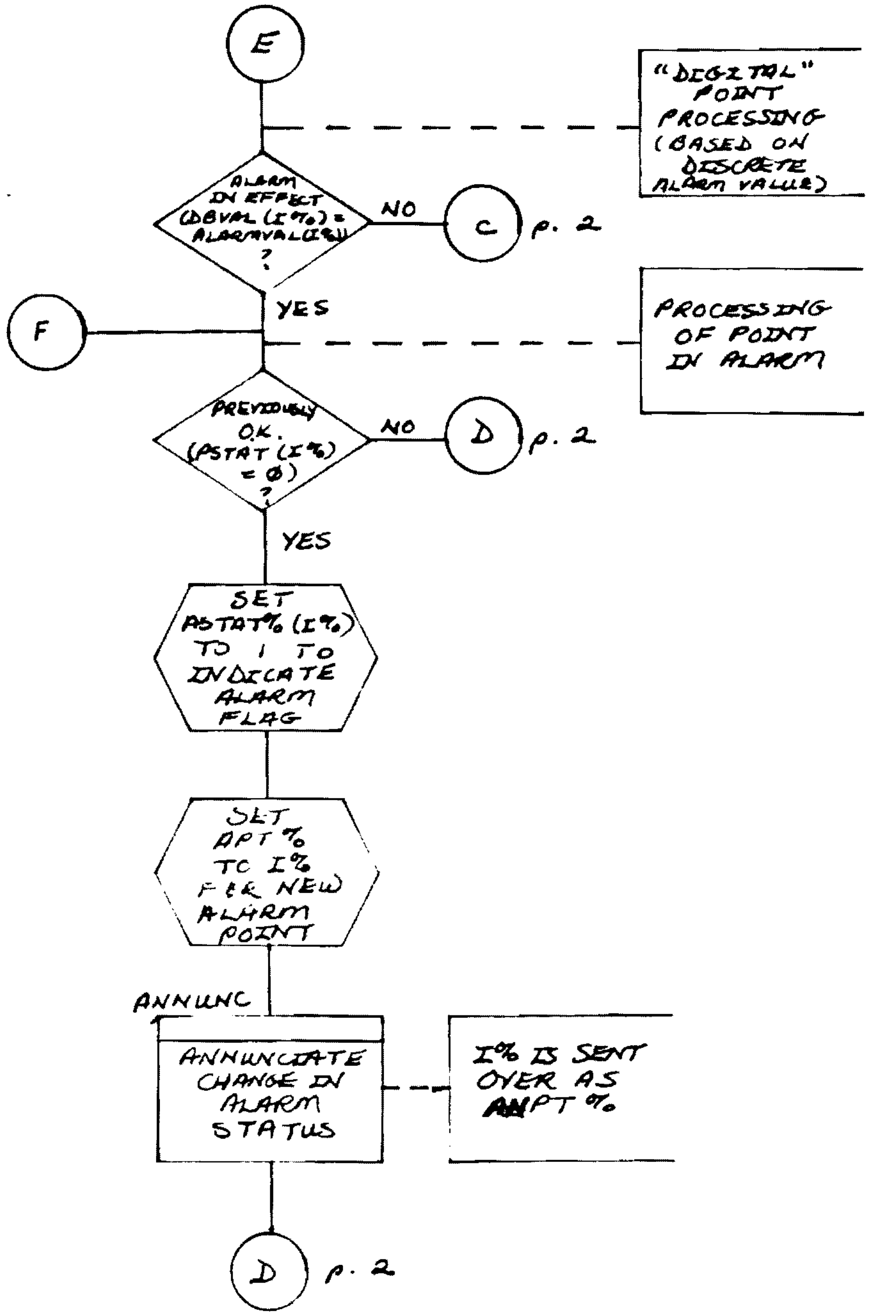
UPDATE: GET REAL-TIME DATABASE VALUES AND CHECK FOR ALARMS

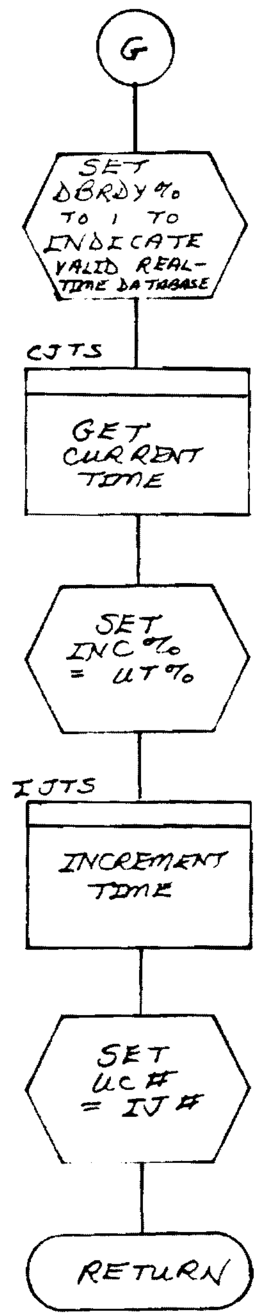


UPDATE: GET REAL-TIME DATABASE VALUES AND CHECK FOR ALARMS



UPDATE: GET REAL-TIME DATABASE VALUES AND CHECK FOR ALARMS




G
SET DBRDY% TO 1 TO INDICATE VALID REAL-TIME DATABASE
CJTS
GET CURRENT TIME
SET INC% = UT%
IJTS
INCREMENT TIME
SET UC# = IJ#
RETURN

NAME: TIME

PURPOSE:

The TIME module is used to initially provide the current time and date to the system time and date software.

OPERATIONAL DESCRIPTION:

The TIME module requests the operator to type in values for the current time and date. From this data, the module computes the Julian date and stores the time information in specific memory locations. It then loads the assembly level clock subroutine from disk and causes it to begin execution.

CALLED BY:

MMI EXEC

CALLS:

Clock - Assembly level clock initialization program

PASSED ARGUMENTS:

JDAY% - Julian day matrix

NAME: TIME (continued)

RETURNED ARGUMENTS:

CJT#	-	Current Julian Time
D%	-	Numerical value of date
H%	-	Numerical value of hour
M%	-	Numerical valuee of minute
M1%	-	Leap year indicator flag
M2%	-	Numberical value of the day of the month
MT%	-	Numerical value of the month
S%	-	Numerical value of the seconds
TC#	-	Time control variable for synchronization of alarm tones
UC#	-	Real-time update synchronization variable
Y%	-	Numerical value of the year

FILE INPUT/OUTPUT:

Retrieves file CLOCK.ABS from disk and stores it in memory sequentially beginning at location 4300 HEX.

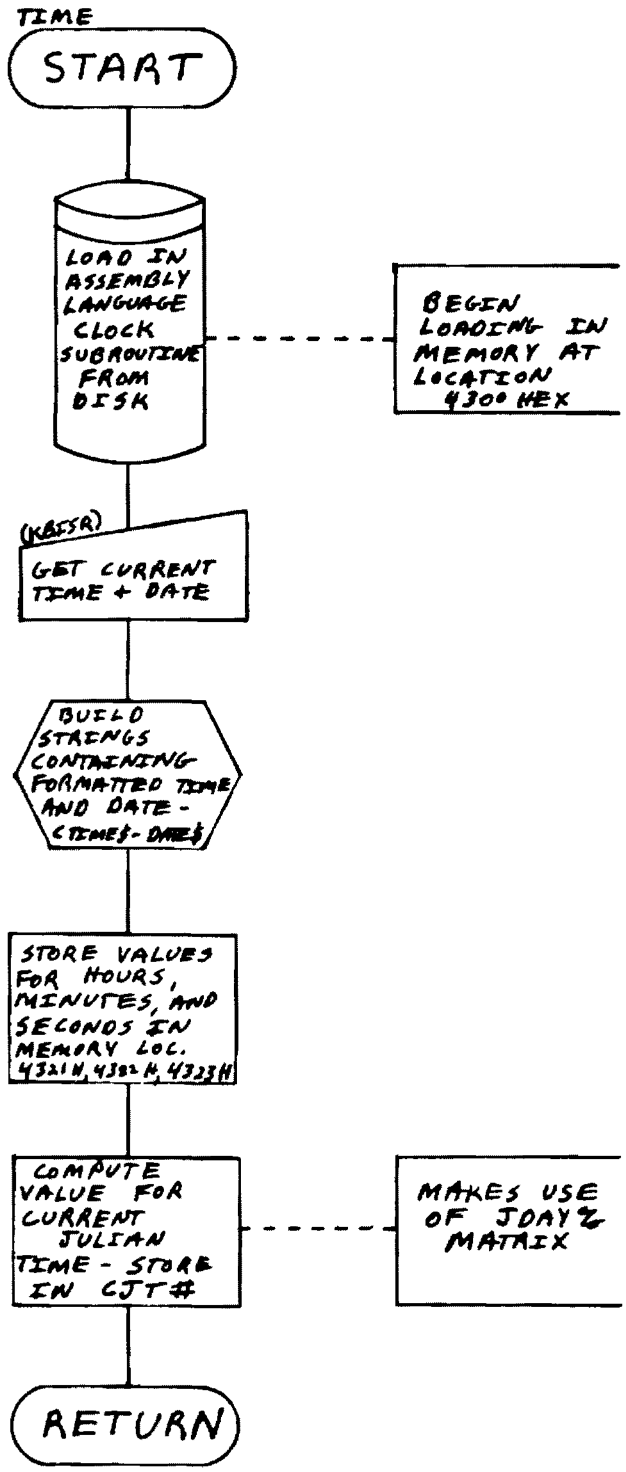
HARDWARE INTERACTION:

Stores data into memory location (hex) 4321, 4322, 4323

CRT Screen - prints messages

DESIGN NOTES:

Julian time is used as a means of providing for an absolute time reference.


TIME
START
LOAD IN ASSEMBLY LANGUAGE CLOCK SUBROUTINE FROM DISK
BEGIN LOADING IN MEMORY AT LOCATION 4300 HEX
(KBISR)
GET CURRENT TIME + DATE
BUILD STRINGS CONTAINING FORMATTED TIME AND DATE - CTIME$-DATE$
STORE VALUES FOR HOURS, MINUTES, AND SECONDS IN MEMORY LOC. 4321H, 4322H, 4323H
COMPUTE VALUE FOR CURRENT JULIAN TIME - STORE IN CJT#
MAKES USE OF JDAY% MATRIX
RETURN

NAME: TOUCHIN

PURPOSE:

The TOUCHIN subroutine is used to get the X & Y coordinates of the point touched on the screen and to call the appropriate subroutine to process the touch request.

OPERATIONAL DESCRIPTION:

The subroutine first obtains the X & Y coordinates of the touch point. It then uses the Y-coordinate to determine which area of the screen was touched, (i.e., alarm indicator, graphics, or function keys). Based on this value and the value of GDTYPE% (which defines whether a menu or process diagram is currently being displayed), the appropriate subroutines are called to process the touch request.

CALLED BY:

MMI EXEC

CALLS:

BUZOFF
FKEYIN
MENUIN
SYMBIN
DISMESS

PASSED ARGUMENTS:

GDTYPE%	- Graphics display type indicator
CAL%(1-8)	- Touch panel calibration factors

NAME: TOUCHIN (continued)

RETURNED ARGUMENTS:

XTUCH%	-	X-coordinate of touched point
YTUCH%	-	Y-coordinate of touched point

FILE INPUT/OUTPUT:

(none)

HARDWARE INTERACTION:

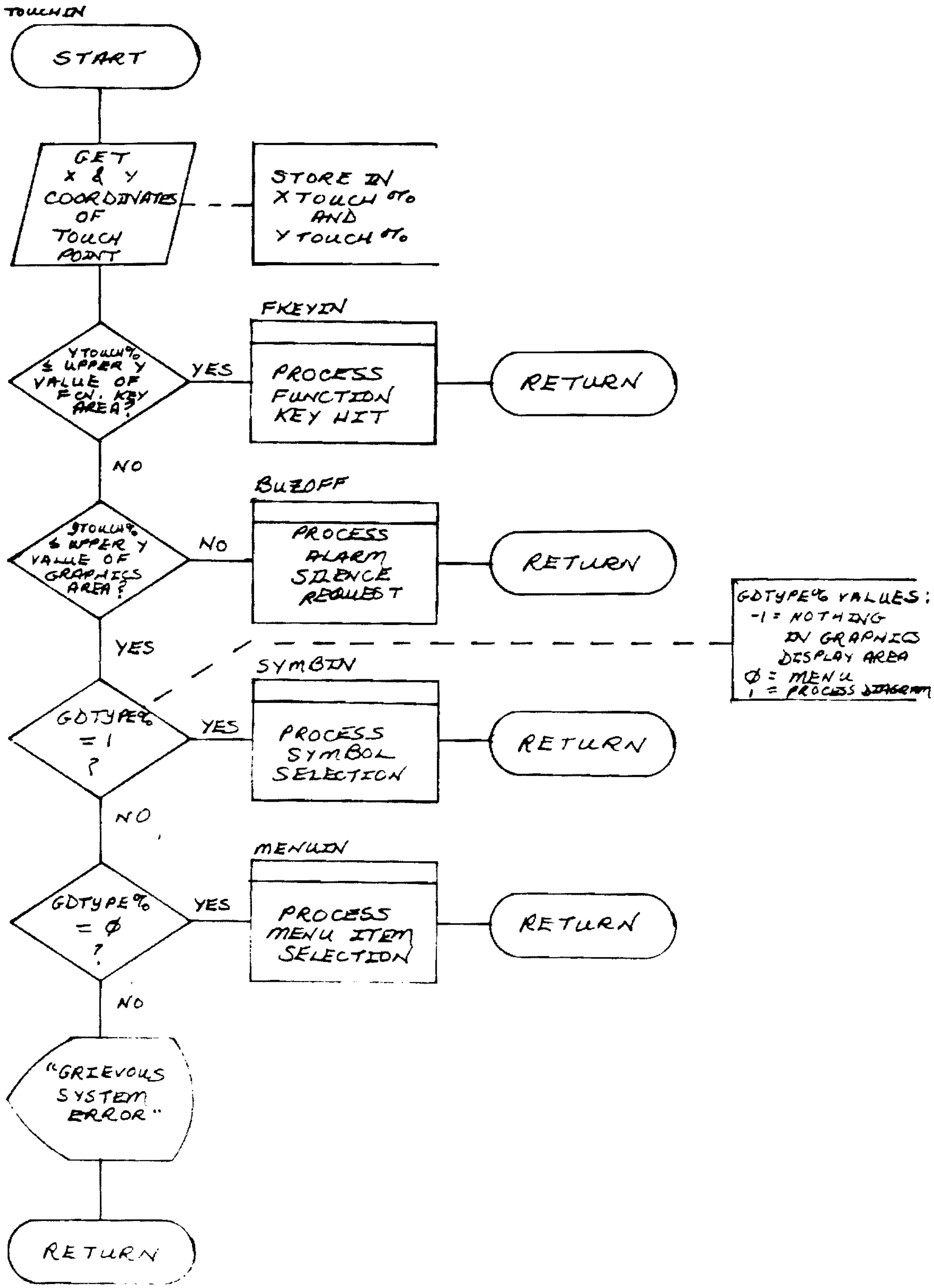
Touch Panel - obtain X & Y coordinates of "hit"

(Probes memory for these values since they are supplied via the TPDRIVER.)

DESIGN NOTES:

(none)

TOUCHIN: GET TOUCH PANEL INPUT



NAME: ANNUNC

PURPOSE:

The ANNUNC subroutine is used to make the screen display reflect any change in alarm status, whether it be to signal a new alarm or to turn the alarm off for a device.

OPERATIONAL DESCRIPTION:

The ANNUNC subroutine first decides whether the point is returning from or entering an alarm condition. If the point is returning from alarm, it sets the appropriate flags to turn off the flashing alarm for the device and also turns off the alarm indicator and buzzer. If the point is entering the alarm state, it starts the buzzer and sets the appropriate flags to cause the alarming device's symbol to flash. It also turns on the alarm indicator.

CALLED BY:

UPDATE

CALLS:

DISMESS
DRAWIT

PASSED ARGUMENTS:

ANPT%	- Annunciation point
APT%	- Current alarm point
ASTAT%	- Alarm status flag
CDBVL	- Current database value for Ith point
DIAMEN%	- Index number of currently displayed diagram or menu
GDTYPE%	- Graphics display type indicator

NAME: ANNUNC (continued)

PASSED ARGUMENTS: (continued)

LPNT%	-	Index number of process diagram on which Ith point is located.
PNAME$	-	String array containing textual name of Ith point
SSED%	-	Start/Stop - Enable/Disable device discriminator array

RETURNED ARGUMENTS:

APT%	-	Current alarm point
SALARM%	-	Flag to determine whether or not to sound the alarm tone
PAUSEALARM#	-	Pause counter for alarm tone silencer
TC#	-	Time control variable for synchronization of alarm tones

FILE INPUT/OUTPUT:

(none)

HARDWARE INTERACTION:

CRT DISPLAY - prints messages

DESIGN NOTES:

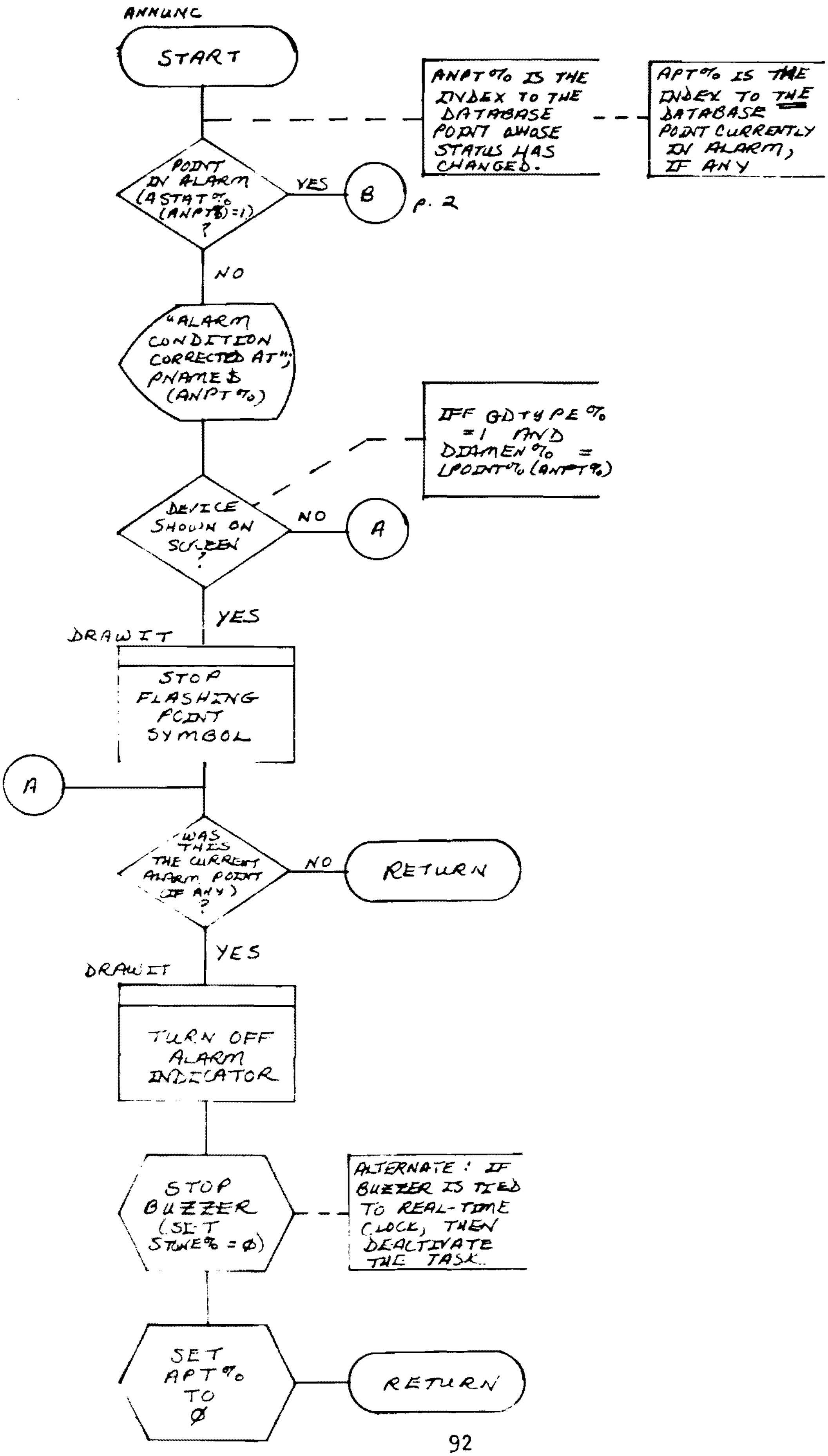
When a change in alarm status occurs, it is necessary to cause the appropriate device symbol to flash or stop flashing. The DISPLAY DIAGRAM command software checks alarm status as it draws a diagram, so points in alarm are always represented as flashing symbols on the screen. However, if the diagram containing the alarm point is already displayed, the ANNUNC subroutine must redraw an alarm point symbol so that it either flashes or stops flashing. The variables GDTYPE%, LPNT%, and DIAMEN% are used by ANNUNC to determine if the diagram containing the alarm point is currently displayed. If GDTYPE% has the value 1, a process diagram is being displayed. DIAMEN% will contain the index number of the diagram

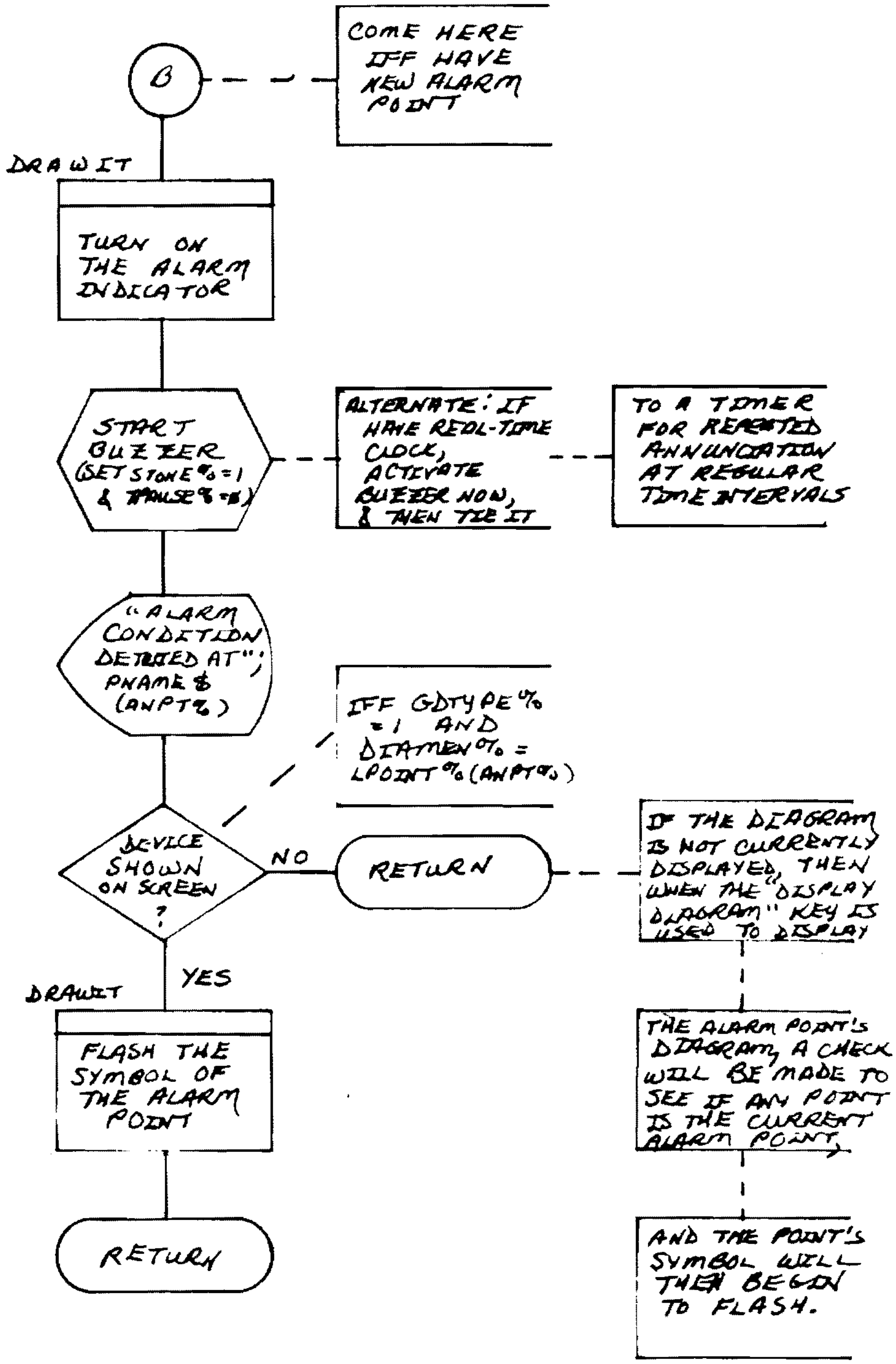
NAME: ANNUNC (concluded)

DESIGN NOTES: (continued)

currently displayed and LPNT% will contain the number of the diagram on which the alarm point is located. If GDTYPE% =1 and DIAMEN%= LPNT%, then ANNUNC knows that it must call the DRAWIT subroutine to cause the symbol of the alarm point to start or stop flashing.

ANNUNC: ANNUNCIATE CHANGE IN ALARM STATUS




B
COME HERE IFF HAVE NEW ALARM POINT
DRAWIT
TURN ON THE ALARM INDICATOR
START BUZZER (SET STONE%=1 & PAUSE%=0)
ALTERNATE: IF HAVE REAL-TIME CLOCK, ACTIVATE BUZZER NOW, & THEN TIE IT
TO A TIMER FOR REPEATED ANNUNCIATION AT REGULAR TIME INTERVALS
"ALARM CONDITION DETECTED AT"; PNAME$ (ANPT%)
IFF GDTYPE% =1 AND DIAMEN% = LPOINT% (ANPT%)
DEVICE SHOWN ON SCREEN ?
NO
RETURN
IF THE DIAGRAM IS NOT CURRENTLY DISPLAYED, THEN WHEN THE "DISPLAY DIAGRAM" KEY IS USED TO DISPLAY
THE ALARM POINT'S DIAGRAM, A CHECK WILL BE MADE TO SEE IF ANY POINT IS THE CURRENT ALARM POINT,
AND THE POINT'S SYMBOL WILL THEN BEGIN TO FLASH.
YES
DRAWIT
FLASH THE SYMBOL OF THE ALARM POINT
RETURN

NAME: FKEYIN

PURPOSE:

The FKEYIN subroutine is used to determine which function key has been selected and to call the appropriate subroutine to process the selected function.

OPERATIONAL DESCRIPTION:

First, a check is made to determine whether or not a function key has already been selected. If so, the old function is deselected. Otherwise the X-coordinate of the touch key is used to determine which key has been touched and to call the appropriate subroutine to process the selected function.

CALLED BY:

TOUCHIN

CALLS:

AUTO
CANCEL
CONFIRM
DIAGRAM
DISMESS
DRAWIT
GONOGO
OPER
REPORT
SETPT
SCHED

NAME: FKEYIN (continued)

PASSED ARGUMENTS:

FKLOC%(I)	-	Leftmost (X) coordinate of Ith function key
QFCN%	-	Previous function key selected
XTUCH%	-	X-coordinate of touch panel input

RETURNED ARGUMENTS:

QFCN%	-	New function key selection

FILE INPUT/OUTPUT:

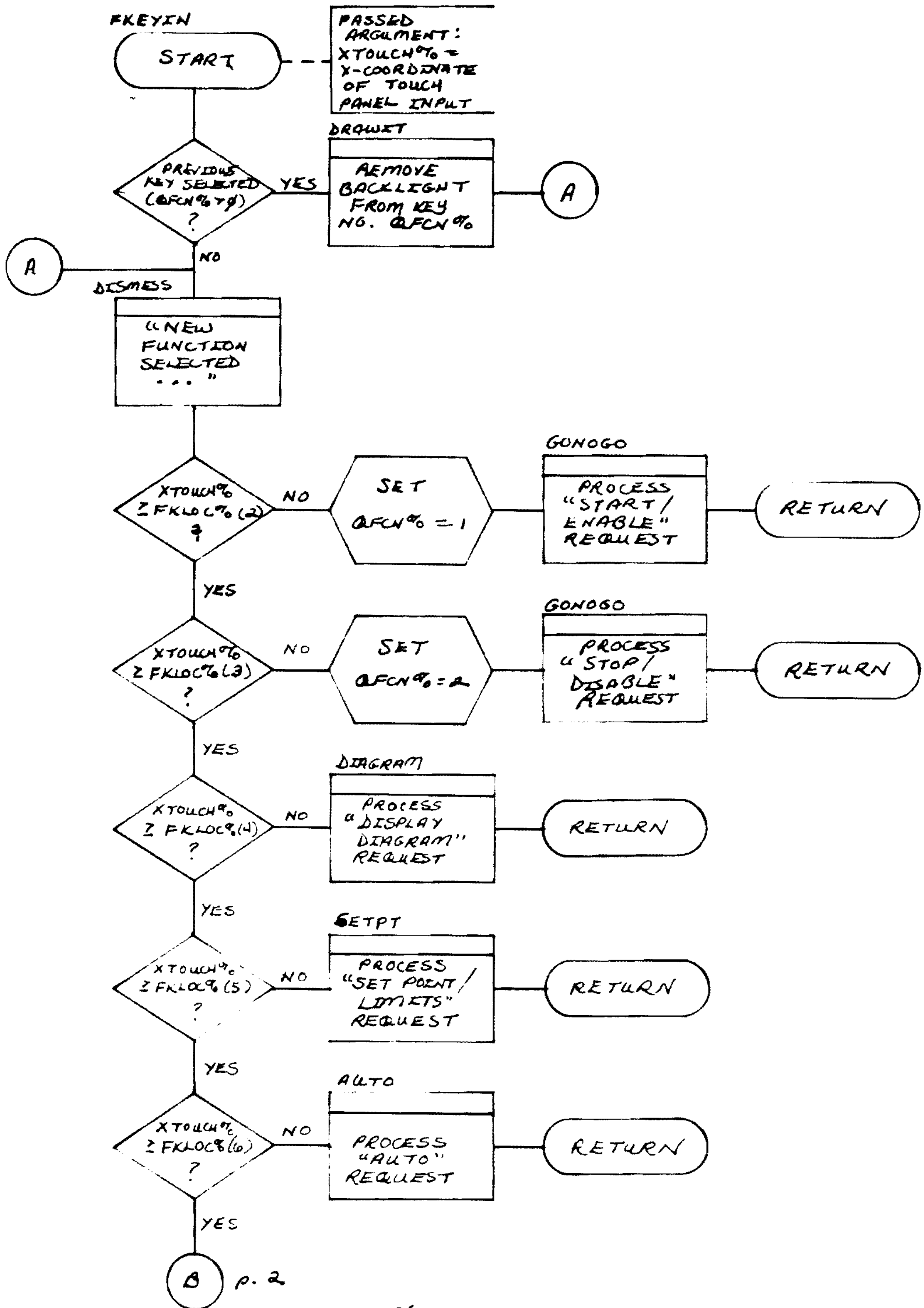
(none)

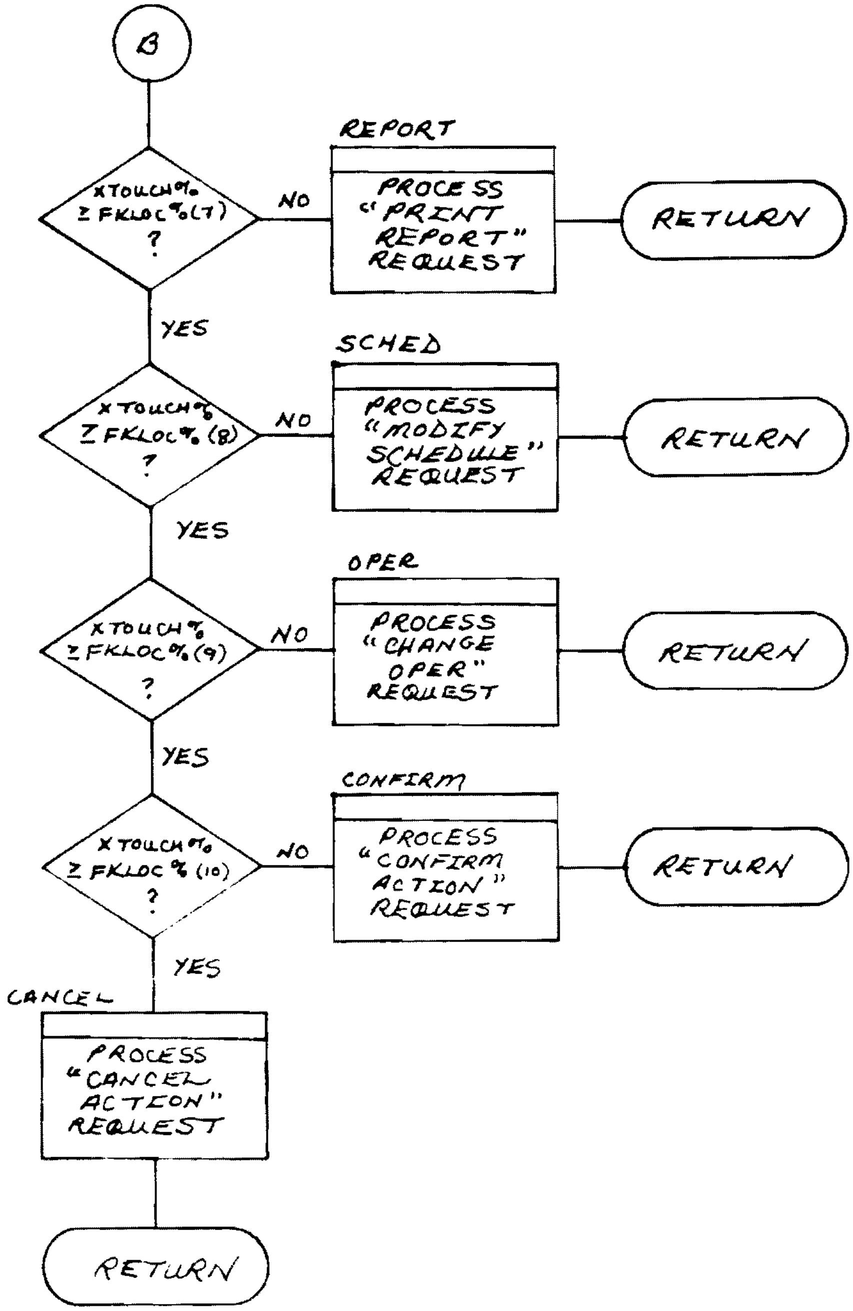
HARDWARE INTERACTION:

(none)

DESIGN NOTES:

(none)


FKEYIN
START
PASSED ARGUMENT: XTOUCH% = X-COORDINATE OF TOUCH PANEL INPUT
PREVIOUS KEY SELECTED (@FCN% ≠ 0) ?
YES
DRQWIT
REMOVE BACKLIGHT FROM KEY NO. @FCN%
A
NO
A
DISMESS
"NEW FUNCTION SELECTED . . ."
XTOUCH% ≥ FKLOC%(2) ?
NO
SET @FCN% = 1
GONOGO
PROCESS "START/ENABLE" REQUEST
RETURN
YES
XTOUCH% ≥ FKLOC%(3) ?
NO
SET @FCN% = 2
GONOGO
PROCESS "STOP/DISABLE" REQUEST
RETURN
YES
XTOUCH% ≥ FKLOC%(4) ?
NO
DIAGRAM
PROCESS "DISPLAY DIAGRAM" REQUEST
RETURN
YES
XTOUCH% ≥ FKLOC%(5) ?
NO
SETPT
PROCESS "SET POINT/LIMITS" REQUEST
RETURN
YES
XTOUCH% ≥ FKLOC%(6) ?
NO
AUTO
PROCESS "AUTO" REQUEST
RETURN
YES
B
p. 2


B
XTOUCH% ≥ FKLOC%(7) ?
NO
YES
REPORT
PROCESS "PRINT REPORT" REQUEST
RETURN
XTOUCH% ≥ FKLOC%(8) ?
NO
YES
SCHED
PROCESS "MODIFY SCHEDULE" REQUEST
RETURN
XTOUCH% ≥ FKLOC%(9) ?
NO
YES
OPER
PROCESS "CHANGE OPER" REQUEST
RETURN
XTOUCH% ≥ FKLOC%(10) ?
NO
YES
CONFIRM
PROCESS "CONFIRM ACTION" REQUEST
RETURN
CANCEL
PROCESS "CANCEL ACTION" REQUEST
RETURN

NAME: SYMBIN

PURPOSE:

The SYMBIN subroutine is used to process device symbol selection requests.

OPERATIONAL DESCRIPTION:

The subroutine first determines which device has been selected and highlights its symbol and displays its status in the text window. If a function has already been selected, checks are made to ensure that conditions are right for execution of the function before control is returned to the calling routine. Appropriate cues and error messages are given.

CALLED BY:

TOUCHIN

CALLS:

DISMESS
DMENU
DRAWIT

PASSED ARGUMENTS:

ALARMVL	- Alarm value for digital point array
ASTAT%	- Alarm status flag for EMCS point array
CDBVL	- Current database value for EMCS point array
DIAMEN%	- Number of the diagram or menu currently displayed
DSTAT%	- Disable status flag
FDEAM%	- Data Environment auto/manual mode indicator
HILIM	- Highest non-alarm value for EMCS analog point array
LALARM%	- Location of alarm (monitor) point types in SID$ array

NAME: SYMBIN (continued)

PASSED ARGUMENTS: (concluded)

LCNT%	- Location of control point types in SID$ array
LIND%	- Location of indicator point types in SID$ array
LOWLIM	- Lowest non-alarm value for EMCS analog point array
LPNT%	- DE on which EMCS point is located array
NALARM%	- Number of monitor (alarm) point types
NCNT%	- Number of control point types
NIND%	- Number of indicator point types
NRTPTS%	- Number of points in real-time data base
PHIER%	- Next higher point in DE hierarchy
PNAME$	- Textual point descriptors
PTYPE%	- Point type indicator
PX%	- Leftmost touch-sensitive X-coordinate for Ith point
PY%	- Lowermost touch-sensitive Y-coordinate for Ith point
QFCN%	- Selected function
QPT%	- Selected point
SSED%	- Start/Stop - Enable/Disable device discriminator array
SPT	- Set point for EMCS analog point array
TT$	- Textual name of data environment
XTUCH%	- X-coordinate of touch "hit"
YTUCH%	- Y-coordinate of touch "hit"

RETURNED ARGUMENTS:

DIAMEN%	- Currently displayed diagram or menu
QPT%	- Selected point

FILE INPUT/OUTPUT:

(none)

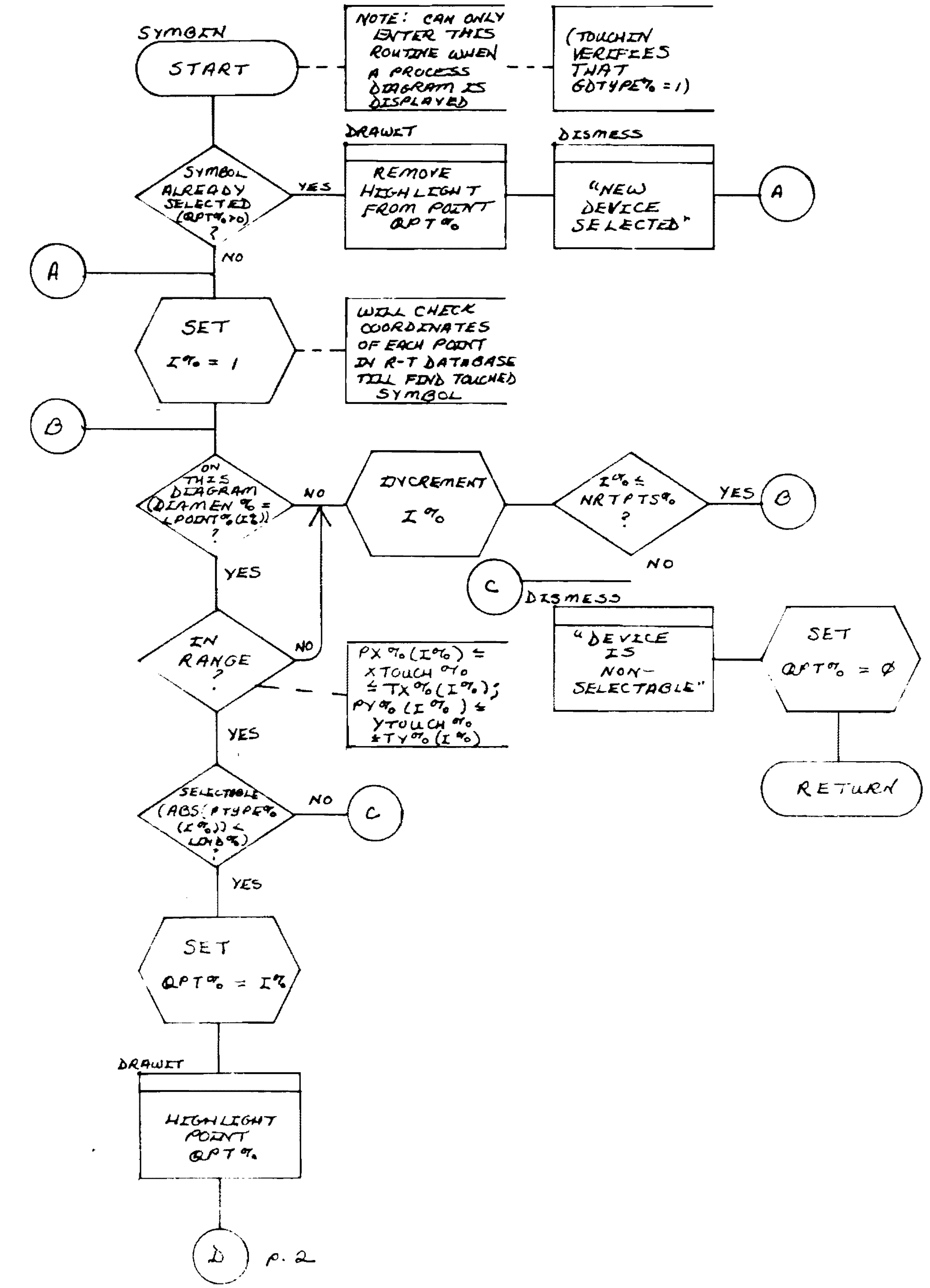
NAME: SYMBIN (concluded)

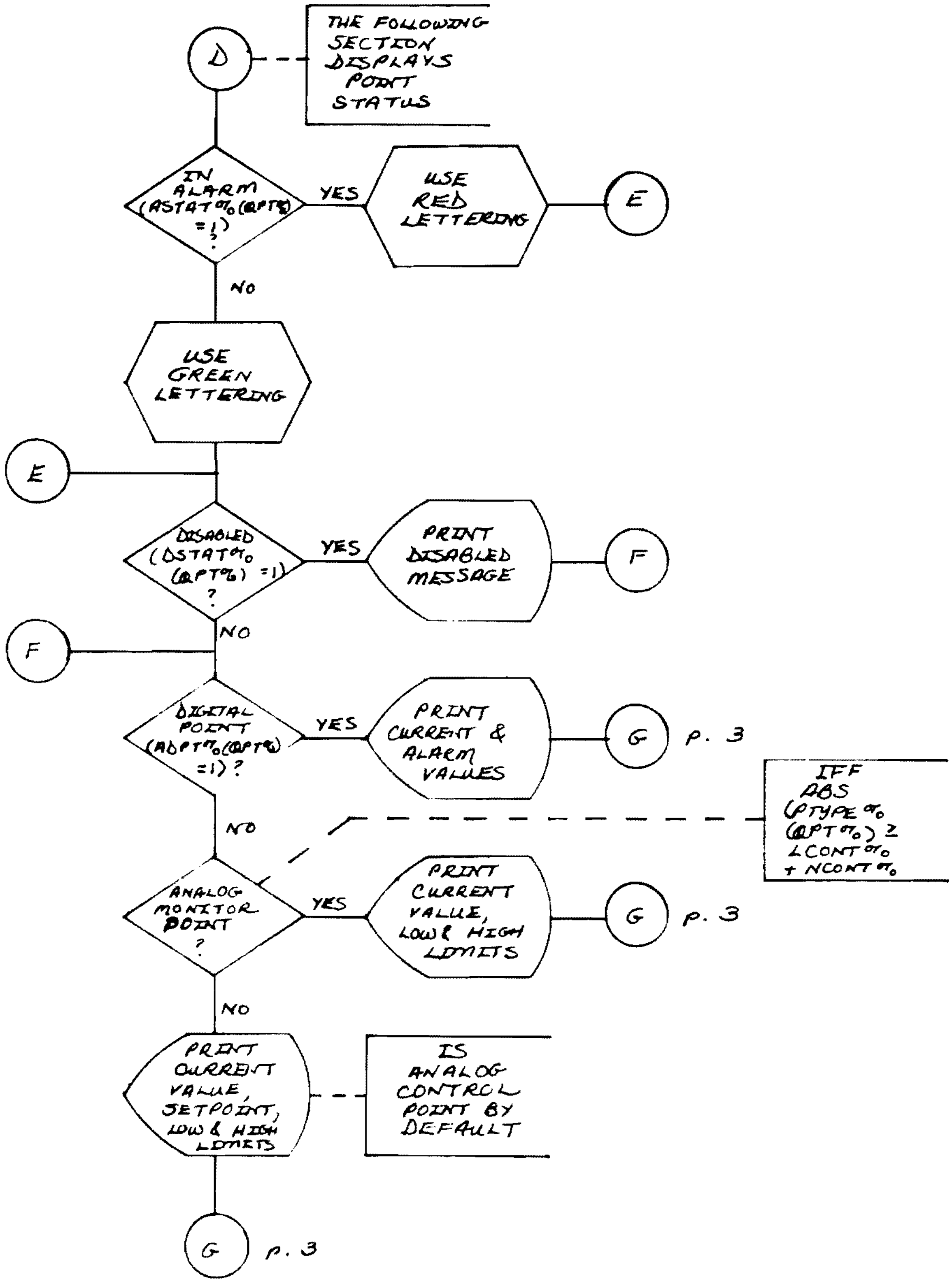
HARDWARE INTERACTION:

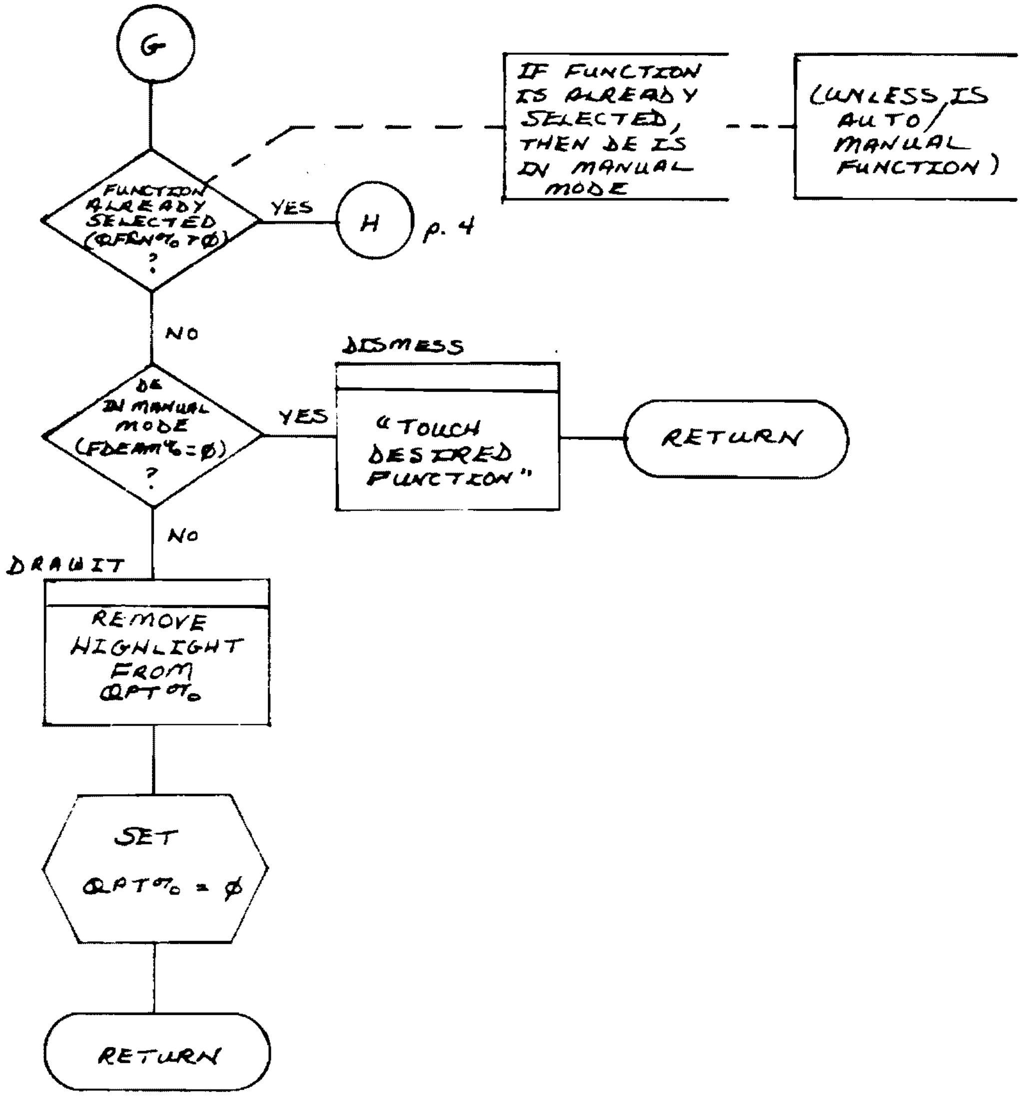
CRT DISPLAY - Display messages

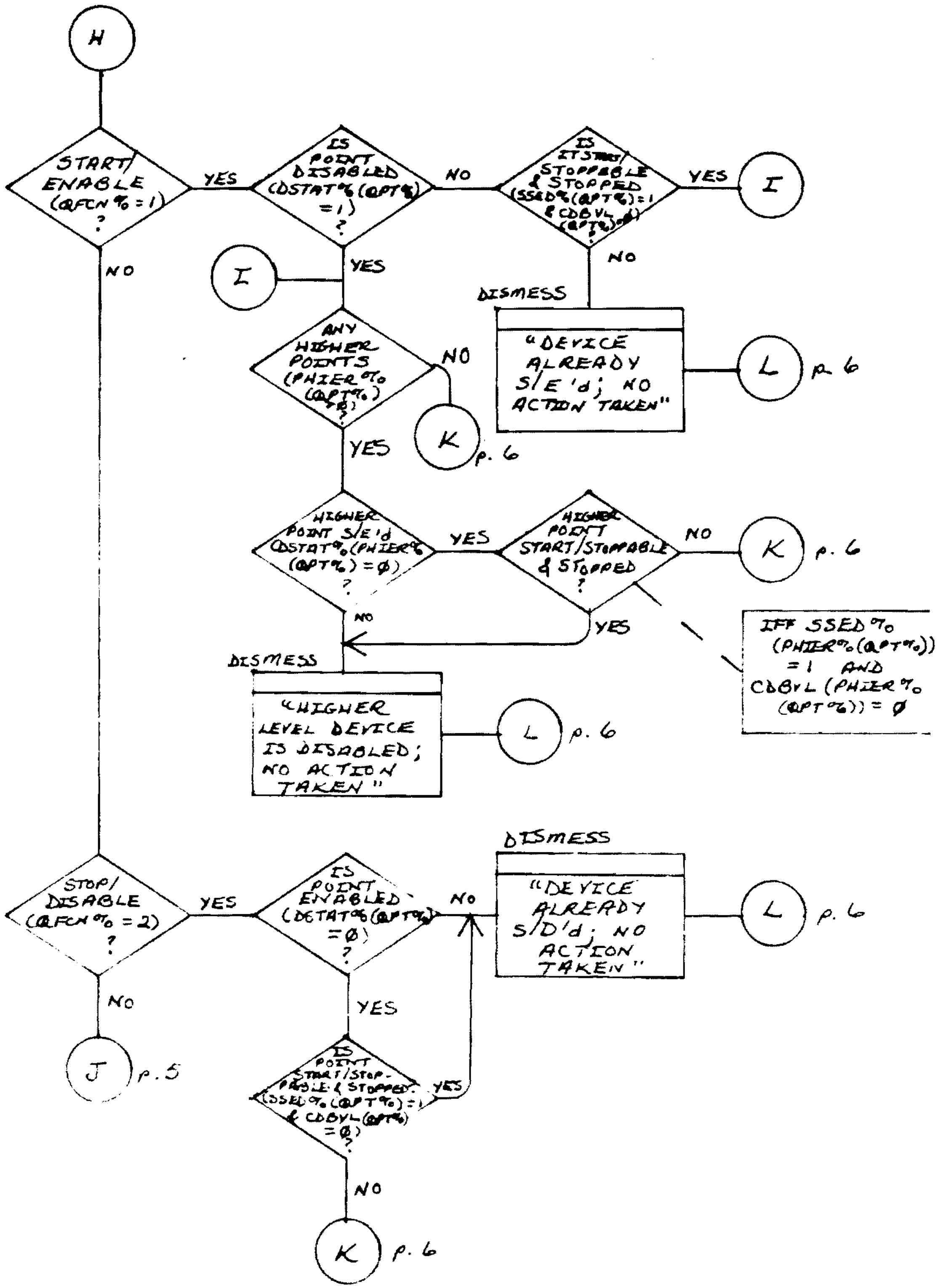
DESIGN NOTES:

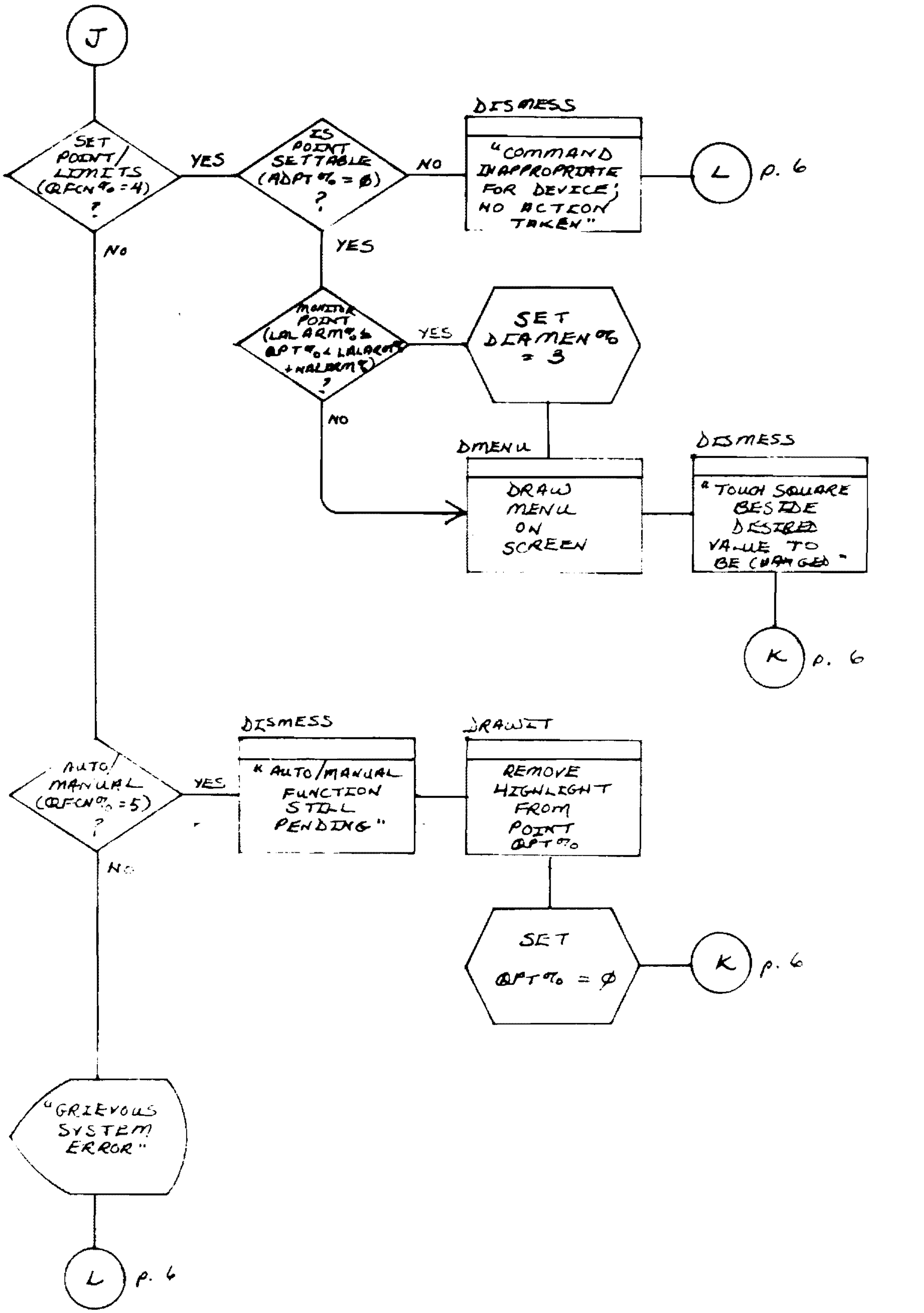
(none)

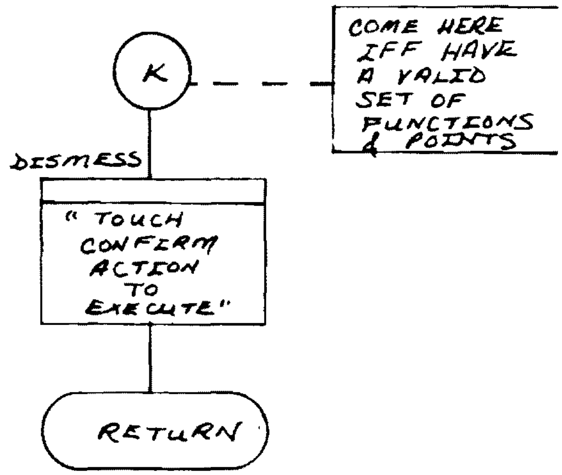

SYMBIN
START
NOTE: CAN ONLY ENTER THIS ROUTINE WHEN A PROCESS DIAGRAM IS DISPLAYED
(TOUCHIN VERIFIES THAT GDTYPE% = 1)
SYMBOL ALREADY SELECTED (QPT%>0) ?
YES
DRAWIT
REMOVE HIGHLIGHT FROM POINT QPT%
DISMESS
"NEW DEVICE SELECTED"
A
NO
A
SET I% = 1
WILL CHECK COORDINATES OF EACH POINT IN R-T DATABASE TILL FIND TOUCHED SYMBOL
B
ON THIS DIAGRAM (DIAMEN% = LPOINT%(I%)) ?
NO
INCREMENT I%
I% ≤ NRTPTS% ?
YES
B
NO
C
DISMESS
"DEVICE IS NON-SELECTABLE"
SET QPT% = Ø
RETURN
YES
IN RANGE ?
NO
PX%(I%) ≤ XTOUCH% ≤ TX%(I%); PY%(I%) ≤ YTOUCH% ≤ TY%(I%)
YES
SELECTABLE (ABS(PTYPE%(I%)) < LOWD%) ?
NO
C
YES
SET QPT% = I%
DRAWIT
HIGHLIGHT POINT QPT%
D
p. 2

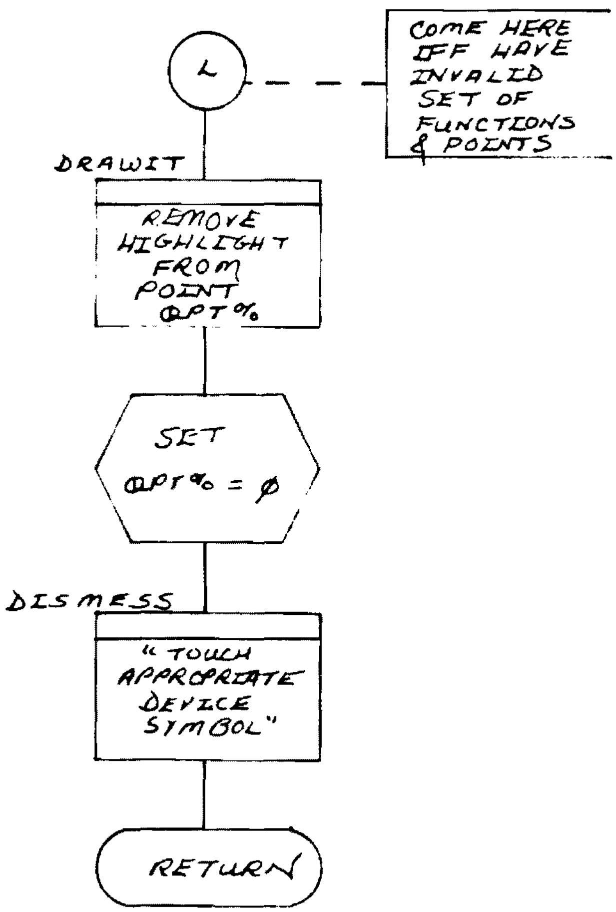

D
THE FOLLOWING SECTION DISPLAYS POINT STATUS
IN ALARM (ASTAT%(@PT%) =1) ?
YES
USE RED LETTERING
E
NO
USE GREEN LETTERING
E
DISABLED (DSTAT% (@PT%) =1) ?
YES
PRINT DISABLED MESSAGE
F
NO
F
DIGITAL POINT (ADPT%(@PT%) =1) ?
YES
PRINT CURRENT & ALARM VALUES
G p. 3
NO
IFF ABS (PTYPE% (@PT%) ≥ LCONT% + NCONT%
ANALOG MONITOR POINT ?
YES
PRINT CURRENT VALUE, LOW & HIGH LIMITS
G p. 3
NO
PRINT CURRENT VALUE, SETPOINT, LOW & HIGH LIMITS
IS ANALOG CONTROL POINT BY DEFAULT
G p. 3


G
FUNCTION ALREADY SELECTED (QFRN% ≠ Ø) ?
YES
H
p. 4
IF FUNCTION IS ALREADY SELECTED, THEN DE IS IN MANUAL MODE
(UNLESS IS AUTO/MANUAL FUNCTION)
NO
DE IN MANUAL MODE (FDEAM% = Ø) ?
YES
DISMESS
"TOUCH DESIRED FUNCTION"
RETURN
NO
DRAWIT
REMOVE HIGHLIGHT FROM QPT%
SET QPT% = Ø
RETURN


H
START/ENABLE (@FCN% = 1) ?
YES
NO
IS POINT DISABLED (DSTAT%(@PT%) = 1) ?
NO
IS IT START/STOPPABLE & STOPPED (SSED%(@PT%)=1 & CDBVL(@PT%)=Ø) ?
YES
I
NO
DISMESS
"DEVICE ALREADY S/E'd; NO ACTION TAKEN"
L p. 6
I
YES
ANY HIGHER POINTS (PHIER%(@PT%) ≠ Ø) ?
NO
K p. 6
YES
HIGHER POINT S/E'd (DSTAT%(PHIER%(@PT%)) = Ø) ?
YES
HIGHER POINT START/STOPPABLE & STOPPED ?
NO
K p. 6
IFF SSED%(PHIER%(@PT%)) = 1 AND CDBVL(PHIER%(@PT%)) = Ø
NO
YES
DISMESS
"HIGHER LEVEL DEVICE IS DISABLED; NO ACTION TAKEN"
L p. 6
STOP/DISABLE (@FCN% = 2) ?
YES
IS POINT ENABLED (DSTAT%(@PT%) = Ø) ?
NO
DISMESS
"DEVICE ALREADY S/D'd; NO ACTION TAKEN"
L p. 6
NO
J p. 5
YES
IS POINT START/STOPPABLE & STOPPED (SSED%(@PT%) = 1 & CDBVL(@PT%) = Ø) ?
YES
NO
K p. 6


J
SET POINT/LIMITS (QFCN% = 4) ?
YES
NO
IS POINT SETTABLE (ADPT% = Ø) ?
NO
YES
DISMESS
"COMMAND INAPPROPRIATE FOR DEVICE; NO ACTION TAKEN"
L p. 6
MONITOR POINT (LALARM% ≤ QPT% ≤ LALARM% + HALARM%) ?
YES
NO
SET DIAMEN% = 3
DMENU
DRAW MENU ON SCREEN
DISMESS
"TOUCH SQUARE BESIDE DESIRED VALUE TO BE CHANGED"
K p. 6
AUTO/MANUAL (QFCN% = 5) ?
YES
NO
DISMESS
"AUTO/MANUAL FUNCTION STILL PENDING"
DRAWIT
REMOVE HIGHLIGHT FROM POINT QPT%
SET QPT% = Ø
K p. 6
"GRIEVOUS SYSTEM ERROR"
L p. 6


K
COME HERE IFF HAVE A VALID SET OF FUNCTIONS & POINTS
DISMESS
" TOUCH CONFIRM ACTION TO EXECUTE"
RETURN


L
COME HERE IFF HAVE INVALID SET OF FUNCTIONS & POINTS
DRAWIT
REMOVE HIGHLIGHT FROM POINT QPT%
SET QPT% = Ø
DISMESS
" TOUCH APPROPRIATE DEVICE SYMBOL"
RETURN

NAME: MENUIN

PURPOSE:

The MENUIN subroutine is used to process menu item selection.

OPERATIONAL DESCRIPTION:

The MENUIN subroutine uses the DIAMEN% variable to determine which menu is to be processed. It then calls the appropriate subroutine to process the particular menu. If a menu choice had been previously made, then the choice is deselected before the call is made.

CALLED BY:

TOUCHIN

CALLS:

DISMESS
DRAWIT
MDIN
MLIN
MOIN
MRIN
MSIN
REPOUT

PASSED ARGUMENTS:

APT%	- Current alarm point
DIAMEN%	- Number of diagram or menu currently displayed
LPNT%	- DE on which EMCS point is located array
QMENU%	- Selected menu item
ZC%	- Internal control variable

NAME: MENUIN (continued)

RETURNED ARGUMENTS:

QMENU%	- Selected menu item
RODS%	- Report output device selection

FILE INPUT/OUTPUT:

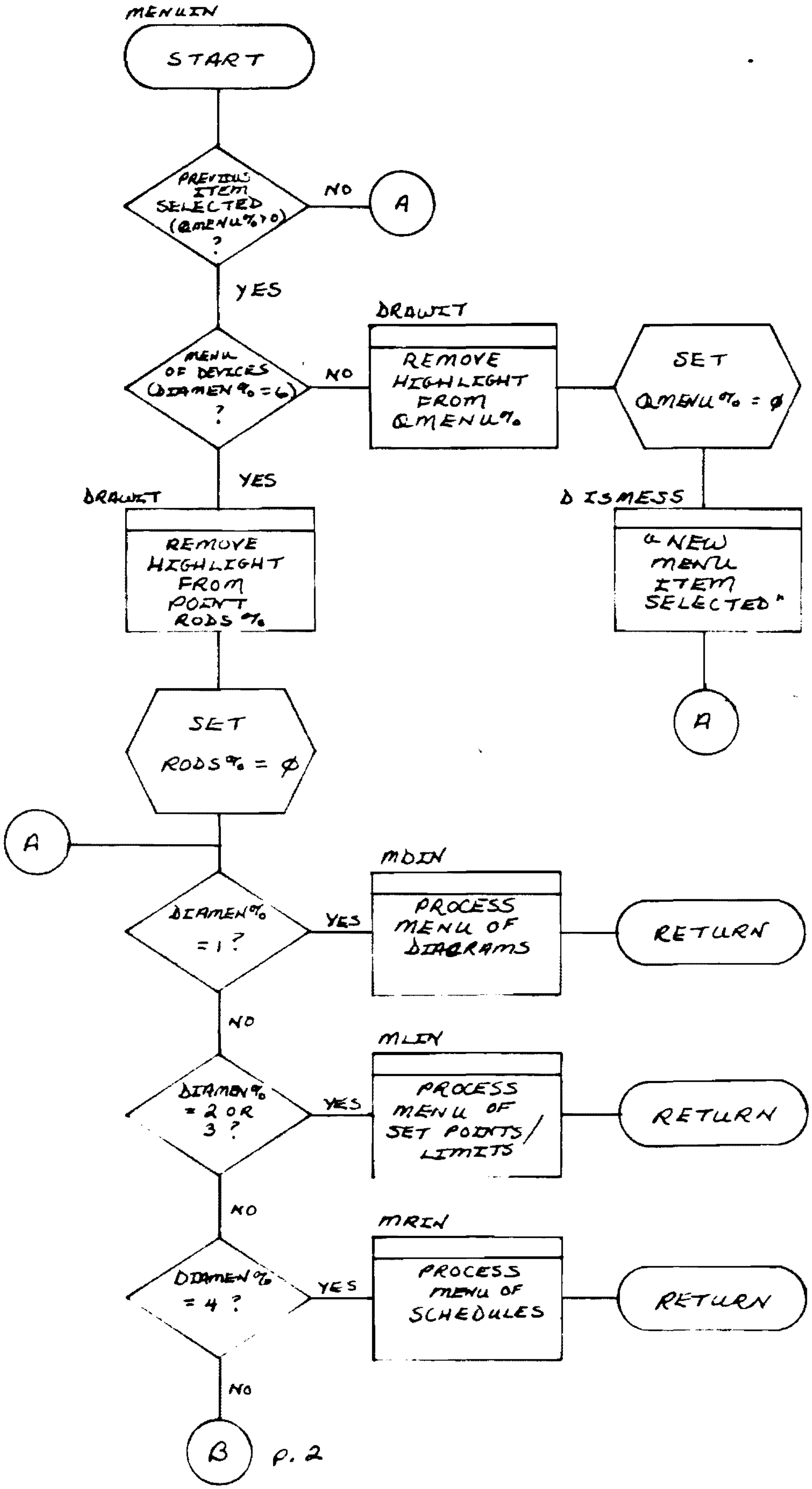
(none)

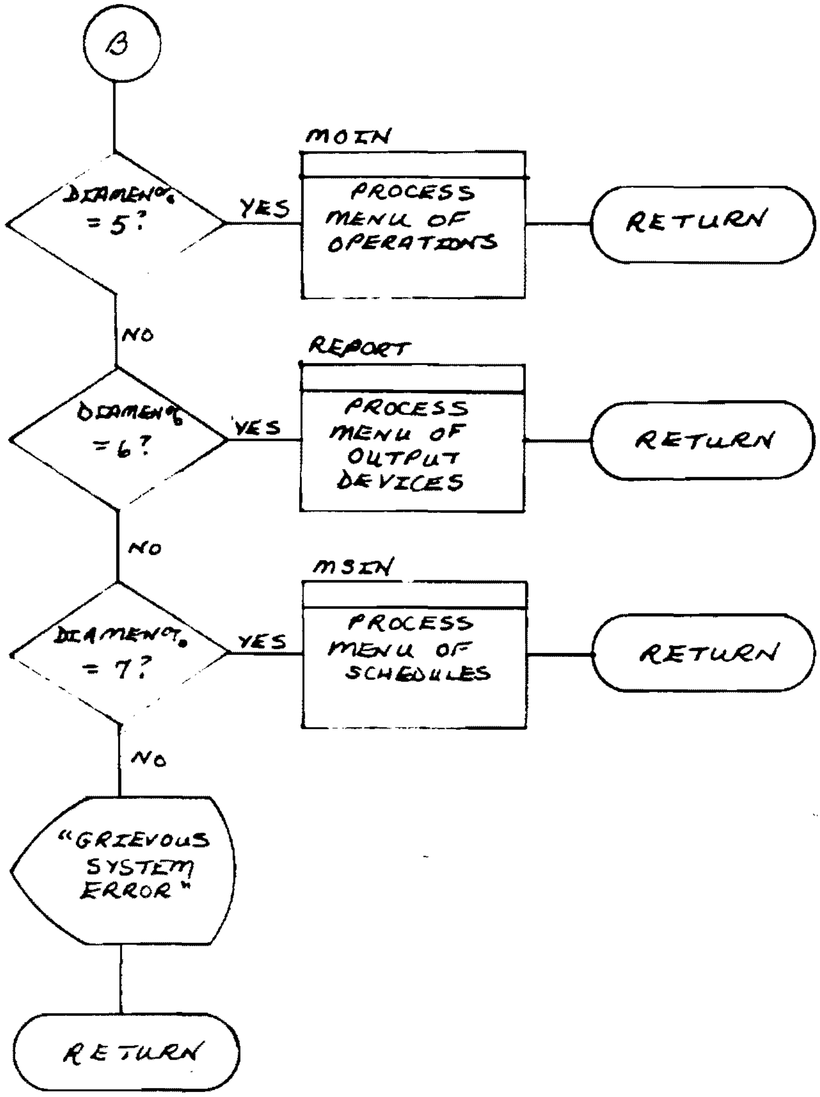
HARDWARE INTERACTION:

(none)

DESIGN NOTES:

(none)


MENUIN
START
PREVIOUS ITEM SELECTED (@MENU%>0) ?
NO
A
YES
DRAWIT
REMOVE HIGHLIGHT FROM @MENU%
MENU OF DEVICES (DIAMEN% = 6) ?
NO
SET @MENU% = ∅
YES
DRAWIT
REMOVE HIGHLIGHT FROM POINT RODS %
DISMESS
"A NEW MENU ITEM SELECTED"
A
SET RODS% = ∅
A
MDIN
DIAMEN% = 1 ?
YES
PROCESS MENU OF DIAGRAMS
RETURN
NO
MLIN
DIAMEN% = 2 OR 3 ?
YES
PROCESS MENU OF SET POINTS/ LIMITS
RETURN
NO
MRIN
DIAMEN% = 4 ?
YES
PROCESS MENU OF SCHEDULES
RETURN
NO
B
p. 2


B
DIAMEN% = 5?
YES
MOIN
PROCESS MENU OF OPERATIONS
RETURN
NO
DIAMEN% = 6?
YES
REPORT
PROCESS MENU OF OUTPUT DEVICES
RETURN
NO
DIAMEN% = 7?
YES
MSIN
PROCESS MENU OF SCHEDULES
RETURN
NO
"GRIEVOUS SYSTEM ERROR"
RETURN

NAME: BUZOFF

PURPOSE:

The purpose of the BUZOFF subroutine is to process the operator alarm silence request.

OPERATIONAL DESCRIPTION:

BUZOFF sets the variable PAUSEALARM# to the current Julian time. It then increments PAUSEALRM# by 120 seconds. The 120 second increment represents the two minute disablement period. The variable TC#, which is used to indicate the time at which the alarm tone was last sounded, is also set equal to PAUSEALRM#. The audible tone will not sound again until the current Julian Time (CJT#) exceeds PAUSEALRM#. BUZOFF displays a message to the operator stating that it has acknowledged the alarm silence request.

CALLED BY:

TOUCHIN

CALLS:

DISMESS
GJTS
IJTS

PASSED ARGUMENTS:

CJT# - Current Julian Time

NAME: BUZOFF (continued)

RETURNED ARGUMENTS:

PAUSEALRM#	- Pause counter for alarm tone silencer
TC#	- Time at which alarm tone last sounded

FILE INPUT/OUTPUT:

(none)

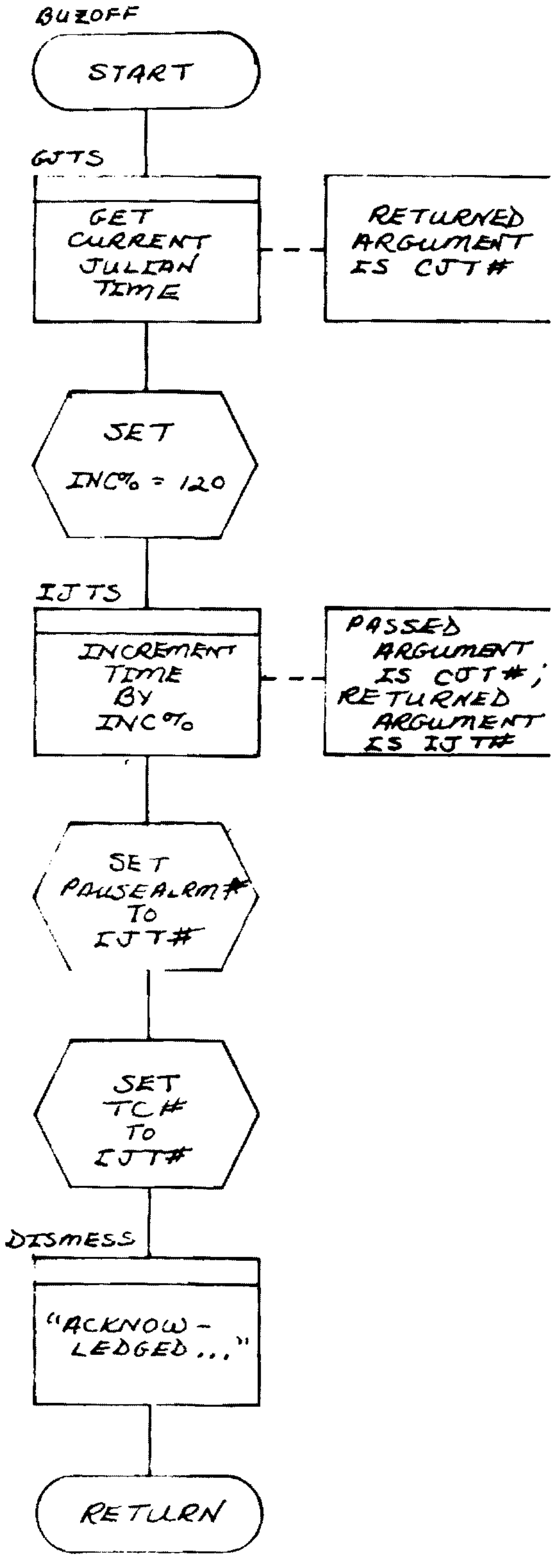
HARDWARE INTERACTION:

(none)

DESIGN NOTES:

The BUZOFF subroutine obtains the current time of the day through GJTS, and IJTS subroutines which interact with an assembly level subroutine which operates the Real Time Clock.

BUZOFF: PROCESS ALARM SILENCE REQUEST



NAME: GONOGO

PURPOSE:

The GONOGO subroutine is used to process the START/ENABLE and STOP/DISABLE command requests. It does not actually execute these functions, but rather ensures that conditions are right for their execution.

OPERATIONAL DESCRIPTION:

The GONOGO subroutine backlights the appropriate function key and checks to make sure that a process diagram is currently displayed. If not, an error message is given. Otherwise, it checks to see if a device has been selected. If not then program control passes to the calling routine. If both a function and device have been selected, a check is made to insure that the point can be operated upon. If so then appropriate cues are given and program control returns to the calling routine. If the function cannot be executed upon the selected point, the point is deselected and appropriate messages are given.

CALLED BY:

FKEYIN

CALLS:

DISMESS
DRAWIT

NAME: GONOGO (continued)

PASSED ARGUMENTS:

CDBVL	- Current database value for Ith point
DSTAT%	- Disable status flag array
FDEAM%	- Auto/Manual flag for current DE
GDTYPE%	- Graphics display type indicator
PHIER%	- Next higher point in process diagram hierarchy
QFCN%	- Function selected
QPT%	- Selected point
SSED%	- Start/Stop - Enable/Disable device discriminator array

RETURNED ARGUMENTS:

QFCN%	- Selected function
QPT%	- Selected point

FILE INPUT/OUTPUT:

(none)

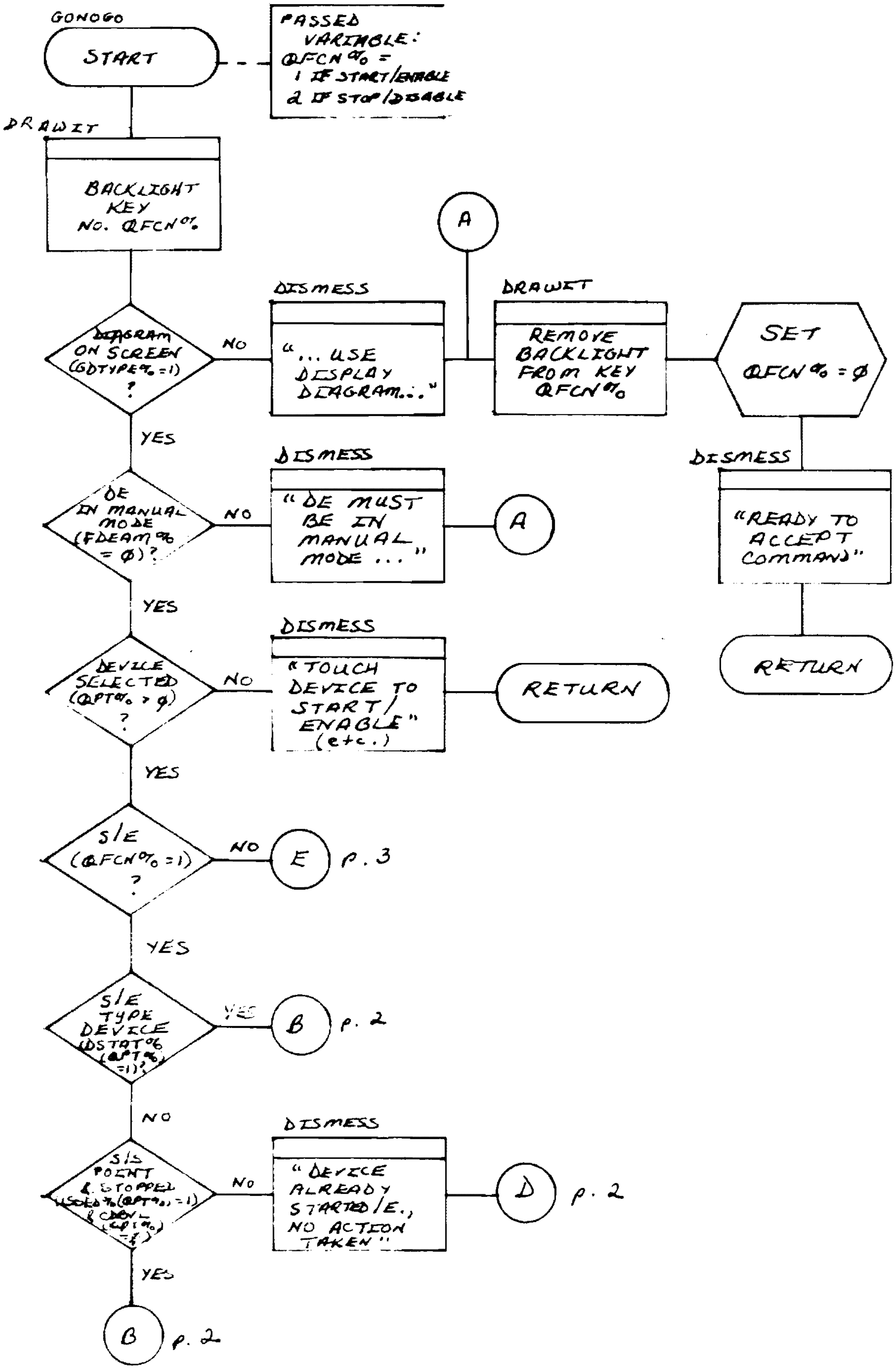
HARDWARE INTERACTION:

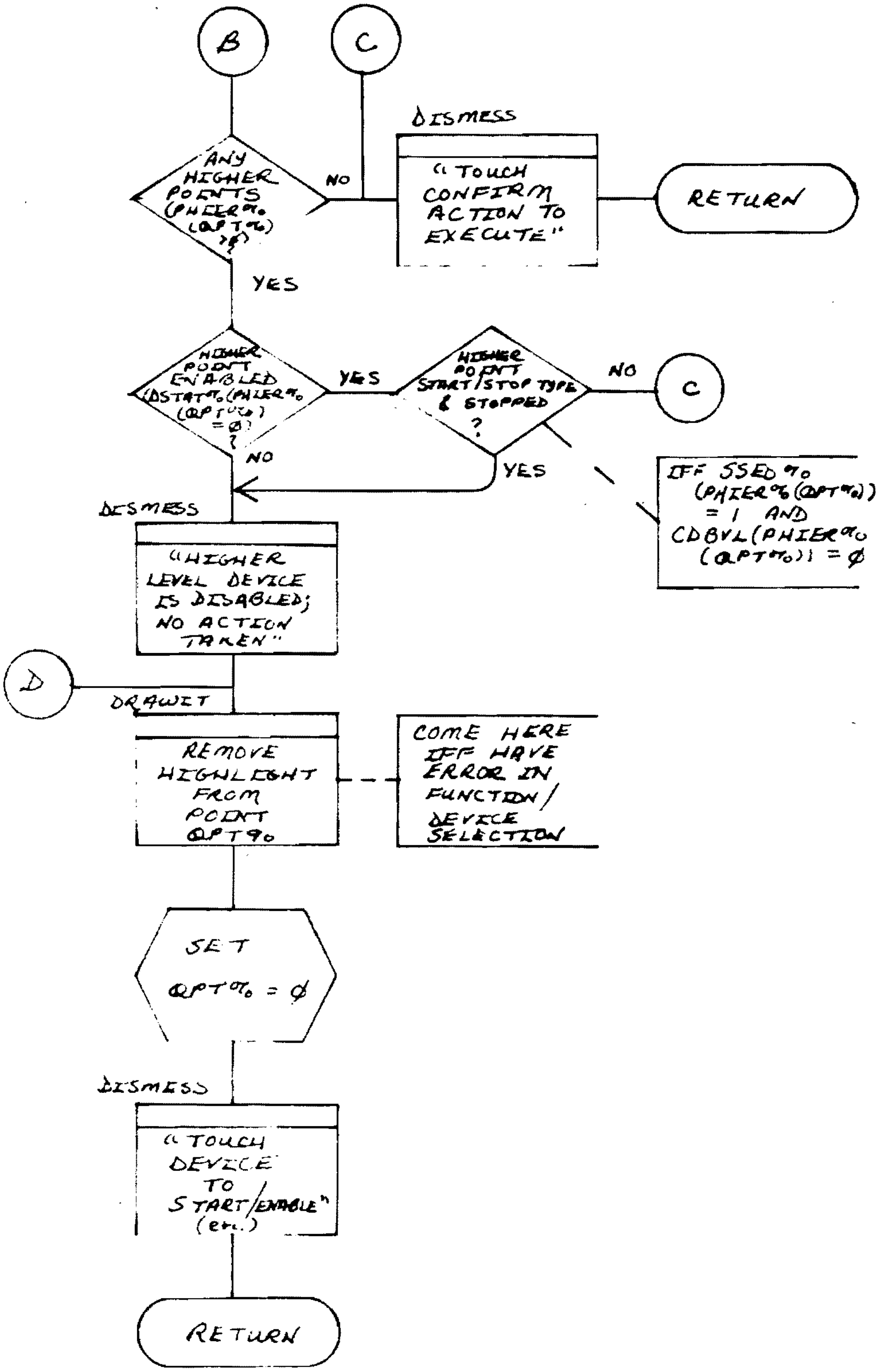
(none)

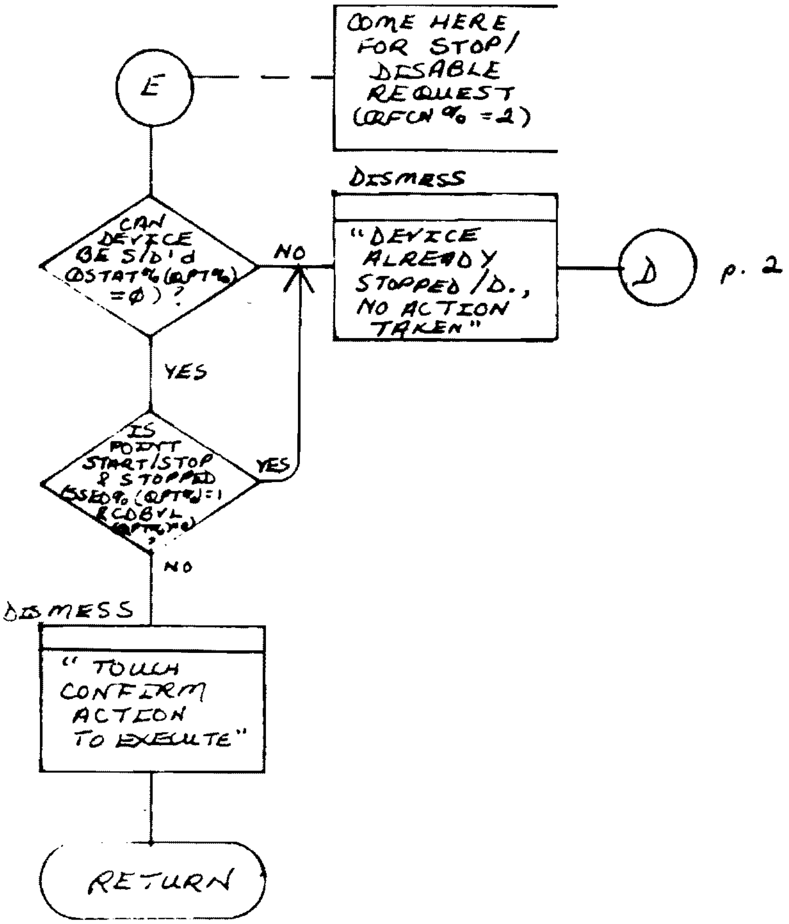
DESIGN NOTES:

(none)

GONOGO: PROCESS "START/ENABLE" OR "STOP/DISABLE" REQUEST




B
C
ANY HIGHER POINTS (PHIER% (QPT%) >Ø) ?
NO
YES
DISMESS
"TOUCH CONFIRM ACTION TO EXECUTE"
RETURN
HIGHER POINT ENABLED (DSTAT%(PHIER%(QPT%)) =Ø) ?
YES
HIGHER POINT START/STOP TYPE & STOPPED ?
NO
C
YES
NO
IFF SSED% (PHIER%(QPT%)) = 1 AND CDBVL(PHIER%(QPT%)) = Ø
DISMESS
"HIGHER LEVEL DEVICE IS DISABLED; NO ACTION TAKEN"
D
DRAWIT
REMOVE HIGHLIGHT FROM POINT QPT%
COME HERE IFF HAVE ERROR IN FUNCTION/DEVICE SELECTION
SET QPT% = Ø
DISMESS
"TOUCH DEVICE TO START/ENABLE" (etc.)
RETURN


E
COME HERE FOR STOP/ DISABLE REQUEST (QRFCN% = 2)
CAN DEVICE BE S/D'd (DSTAT%(QPT%) = Ø)?
NO
YES
DISMESS
"DEVICE ALREADY STOPPED/D., NO ACTION TAKEN"
D
p. 2
YES
NO
DISMESS
"TOUCH CONFIRM ACTION TO EXECUTE"
RETURN

NAME: DIAGRAM

PURPOSE:

The DIAGRAM subroutine is used to process the DISPLAY DIAGRAM command request.

OPERATIONAL DESCRIPTION:

The DISPLAY DIAGRAM function key is backlighted and the appropriate control variables are initialized. The DMENU subroutine is called to display the available DE selections.

CALLED BY:

FKEYIN

CALLS:

DISMESS
DMENU
DRAWIT

PASSED ARGUMENTS:

(none)

RETURNED ARGUMENTS:

DIAMEN% - Number of diagram or menu currently displayed
QFCN% - Selected function
QMENU% - Selected menu item
QPT% - Selected point

NAME: DIAGRAM (continued)

FILE INPUT/OUTPUT:

(none)

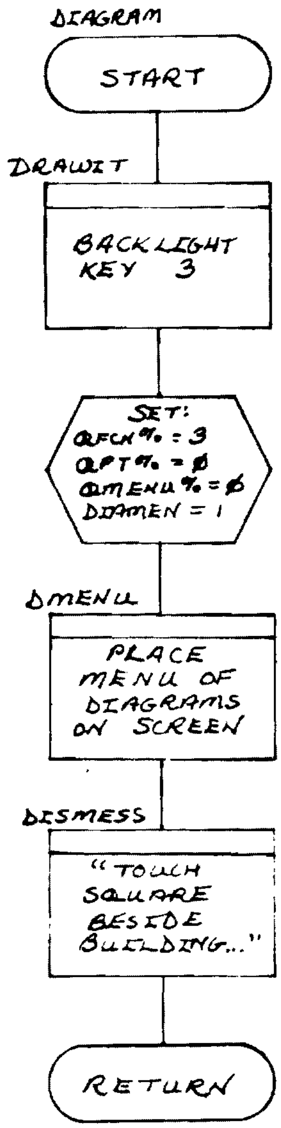
HARDWARE INTERACTITON:

(none)

DESIGN NOTES:

(none)

DIAGRAM: PROCESS "DISPLAY DIAGRAM" REQUEST



NAME: SETPT

PURPOSE:

The SETPT subroutine is used to process the SET POINT/LIMITS command request. SETPT does not actually execute the command, but rather ensures that conditions are right for execution.

OPERATIONAL DESCRIPTION:

The SET POINT/LIMITS function key is backlighted and checks are made to ensure that a process diagram is on the screen and that, if a point has been selected, it is of the appropriate type. Appropriate cues and error messages are given.

CALLED BY:

FKEYIN

CALLS:

DISMESS
DMENU
DRAWIT

PASSED ARGUMENTS:

ADPT%	- Analog/Digital point discriminator array
FDEAM%	- Auto/Manual flag for current DE
GDTYPE%	- Graphics display type indicator
LALARM%	- Location of alarm (monitor) point types in SID$ array
NALARM%	- Number of monitor (alarm) point types
PTYPE%	- Type of point
QPT%	- Selected point

NAME: SETPT (continued)

RETURNED ARGUMENTS:

DIAMEN%	- Number of diagram or menu currently displayed
QFCN%	- Selected function
QMENU%	- Selected menu item
QPT%	- Selected point

FILE INPUT/OUTPUT:

(none)

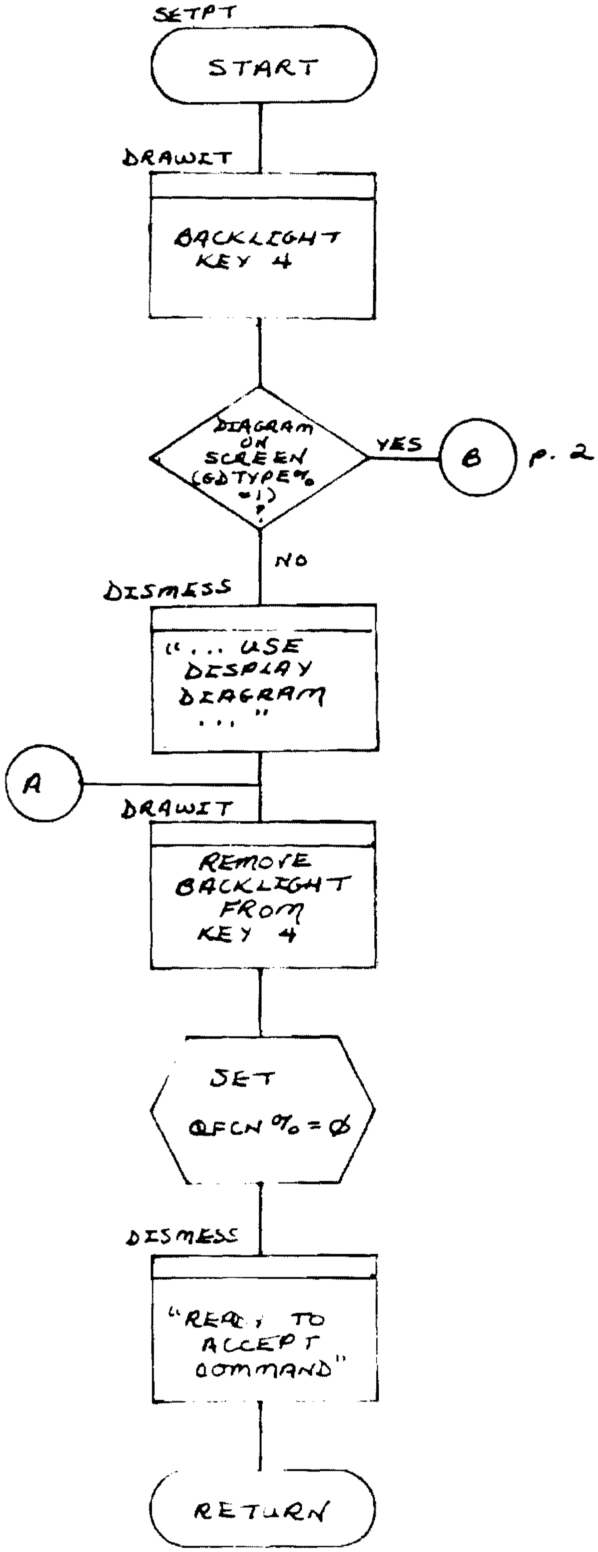
HARDWARE INTERACTION:

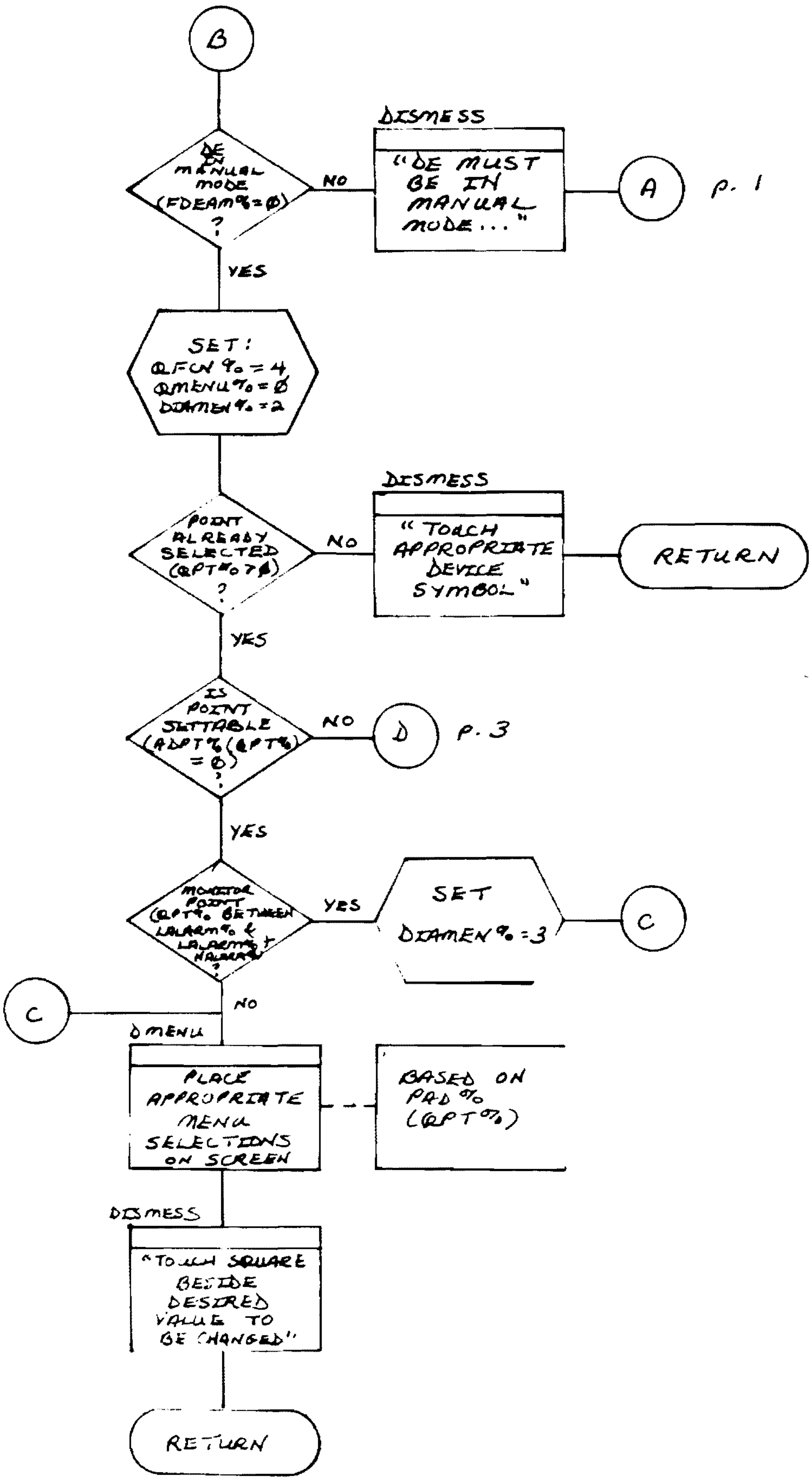
(none)

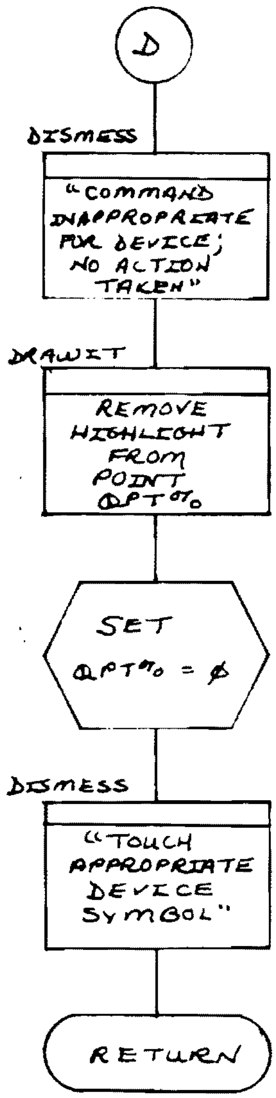
DESIGN NOTES:

(none)

SETPT: PROCESS "SET POINT/LIMITS" REQUEST




B
DE IN MANUAL MODE (FDEAM%=Ø) ?
NO
DISMESS
"DE MUST BE IN MANUAL MODE..."
A
p. 1
YES
SET: QFCN%=4 QMENU%=Ø DIAMEN%=2
POINT ALREADY SELECTED (QPT%>Ø) ?
NO
DISMESS
"TOUCH APPROPRIATE DEVICE SYMBOL"
RETURN
YES
IS POINT SETTABLE (ADPT%(QPT%)=Ø) ?
NO
D
p. 3
YES
MONITOR POINT (QPT% BETWEEN LALARM% & LALARM%+HALARM%) ?
YES
SET DIAMEN%=3
C
NO
C
DMENU
PLACE APPROPRIATE MENU SELECTIONS ON SCREEN
BASED ON PAD% (QPT%)
DISMESS
"TOUCH SQUARE BESIDE DESIRED VALUE TO BE CHANGED"
RETURN


D
DISMESS
"COMMAND INAPPROPRIATE FOR DEVICE; NO ACTION TAKEN"
DRAWIT
REMOVE HIGHLIGHT FROM POINT @PT%
SET @PT% = ∅
DISMESS
"TOUCH APPROPRIATE DEVICE SYMBOL"
RETURN

NAME: AUTO

PURPOSE:

The AUTO subroutine is used to process the AUTO/MANUAL command request. AUTO does not actually execute the command, but rather ensures that all conditions are right for its execution.

OPERATIONAL DESCRIPTION:

The AUTO/MANUAL key is backlighted and checks are made to ensure that a DE diagram is currently displayed. Based on these checks, the appropriate control and selection variable values are set and appropriate cues and error messages are given.

CALLED BY:

FKEYIN

CALLS:

DISMESS
DRAWIT

PASSED ARGUMENTS:

DIAMEN%	- Number of diagram or menu currently displayed
FDEAM%	- Data environment auto/manual indicator
GDTYPE%	- Graphics display type
TT$	- Textual name of data environment

NAME: AUTO (continued)

RETURNED ARGUMENT:

QAM%	- Selected mode
QFCN%	- Selected function

FILE INPUT/OUTPUT:

(none)

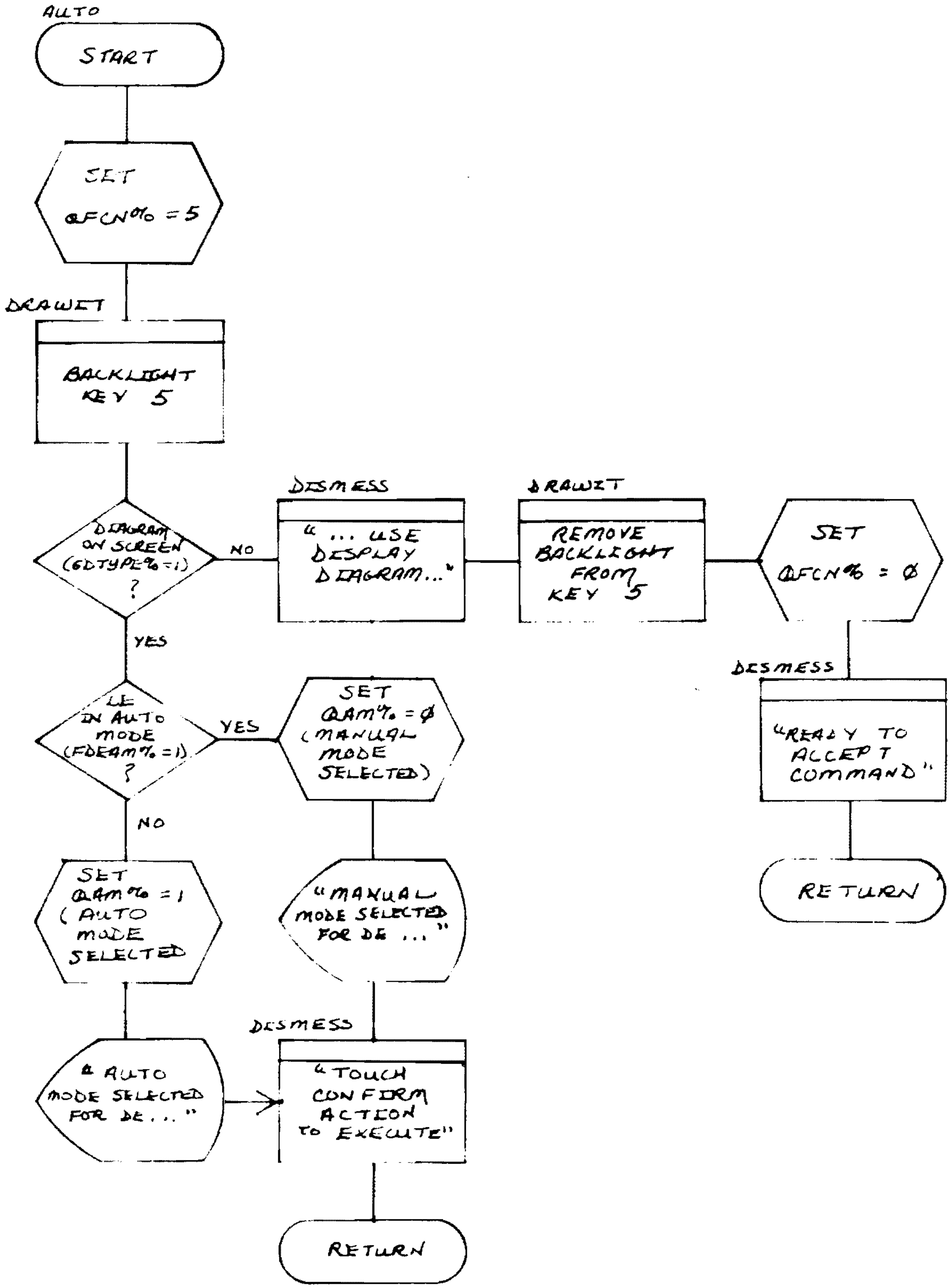
HARDWARE INTERACTION:

CRT Display - Print messages

DESIGN NOTES:

(none)

AUTO: PROCESS "AUTO/MANUAL" REQUEST



NAME: REPORT

PURPOSE:

The REPORT subroutine is used to process the PRINT REPORT command request.

OPERATIONAL DESCRIPTION:

The PRINT REPORT function key is backlighted and the values for the control and selection variables are set. The DMENU subroutine is called to display the menu of available reports.

CALLED BY:

FKEYIN

CALLS:

DISMESS
DMENU
DRAWIT

PASSED ARGUMENTS:

(none)

RETURNED ARGUMENTS:

DIAMEN%	- Number of diagram or menu currently displayed
QFCN%	- Selected function
QMENU%	- Selected menu item
QPT%	- Selected point
RODS%	- Report output device selection

NAME: REPORT (continued)

FILE INPUT/OUTPUT:

(none)

HARDWARE INTERACTION:

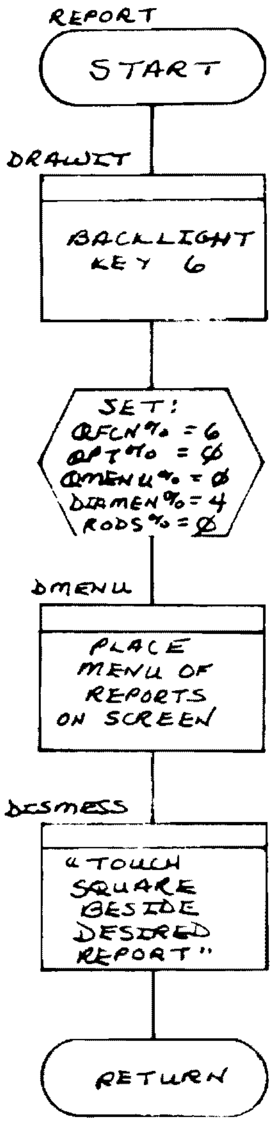
(none)

DESIGN NOTES:

(none)

REPORT: PROCESS "PRINT REPORT" REQUEST

page 1 of 1



NAME: SCHED

PURPOSE:

The SCHED subroutine is used to process the MODIFY SCHED command request.

OPERATIONAL DESCRIPTION:

The MODIFY SCHED function key is backlighted. Selection and control variables are set and DMENU is called to display the menu of schedules on the screen.

CALLED BY:

FKEYIN

CALLS:

DISMESS
DMENU
DRAWIT

PASSED ARGUMENTS:

RETURNED ARGUMENTS:

DIAMEN%	- Number of diagram or menu currently displayed
QFCN%	- Selected function
QMENU%	- Selected menu item
QSCH%	- Selected data environment schedule
S$(QSCH%)	- Selected data environment schedule values

NAME: SCHED (continued)

FILE INPUT/OUTPUT:

(none)

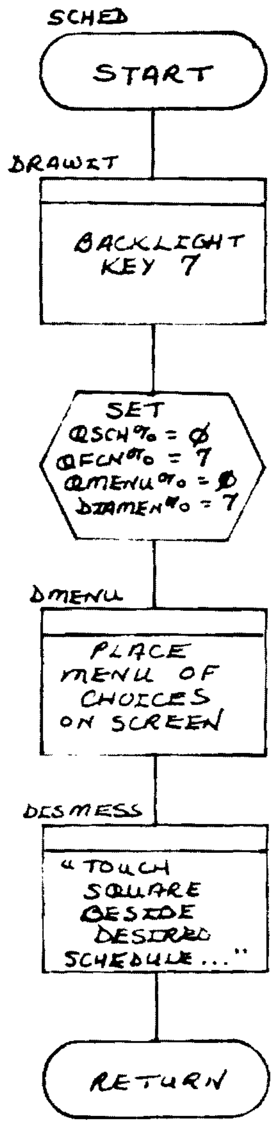
HARDWARE INTERACTION:

(none)

DESIGN NOTES:

(none)

SCHED: PROCESS "MODIFY SCHED" REQUEST



NAME: OPER

PURPOSE:

The OPER subroutine is used to process the CHANGE OPER request.

OPERATIONAL DESCRIPTION:

The CHANGE OPER function key is backlighted and the values for the control and selection variables are set. The DMENU subroutine is called to display the menu of available operations.

CALLED BY:

FKEYIN

CALLS:

DISMESS

DMENU

DRAWIT

PASSED ARGUMENTS:

(none)

RETURNED ARGUMENTS:

DIAMEN%	- Currently displayed diagram
QAM%	- Selected mode
QFCN%	- Selected function
QMENU%	- Selected menu item
QOPER%	- Current operator
QPT%	- Selected point

NAME: OPER (continued)

FILE INPUT/OUTPUT:

(none)

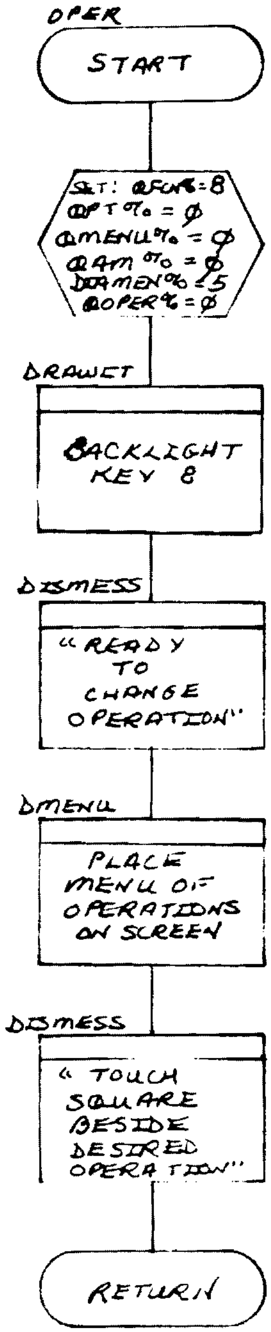
HARDWARE INTERACTION:

(none)

DESIGN NOTES:

(none)

OPER: PROCESS CHANGE OPERATION REQUEST



NAME: CONFIRM

PURPOSE:

The CONFIRM subroutine is used to process the CONFIRM ACTION command request.

OPERATIONAL DESCRIPTION:

The CONFIRM ACTION key is backlighted and checks are made to ensure that a function and all its parameters have been selected. Appropriate error messages are given for improper or incomplete selections. If all selections have been made, then based upon which function has been selected, the appropriate subroutine is called to execute the function. Upon completion of the function execution, CONFIRM restores the graphics area to its previous state, (i.e., if a menu had overwritten the DE diagram then CONFIRM would cause the DE to be redisplayed.)

CALLED BY:

FKEYIN

CALLS:

DISMESS
DMENU
DRAWIT
DSCHED
REPOUT
XAM
XCHOP
XDD
XMS
XPR
XSD
XSE
XSPL

NAME: CONFIRM (continued)

PASSED ARGUMENTS:

DIAMEN%	-	Currently displayed diagram or menu
GDTYPE%	-	Graphics display type
LDE%	-	Number of the last DE displayed
NDES%	-	Number of data environments in system
NUMPRTS%	-	Number of available reports
QAM%	-	Selected mode
QFCN%	-	Selected function
QMENU%	-	Selected menu item
QPT%	-	Selected point
QSCH%	-	Selected schedule
RODS%	-	Report output device selection
SHUTDOWN%	-	Flag to stop system
ZC%	-	Internal control variable

RETURNED ARGUMENTS:

DIAMEN%	-	Currently displayed diagram or menu
GDTYPE%	-	Graphics display type
QFCN%	-	Selected function
QMENU%	-	Selected menu item
ZC%	-	Internal control variable

FILE INPUT/OUTPUT:

(none)

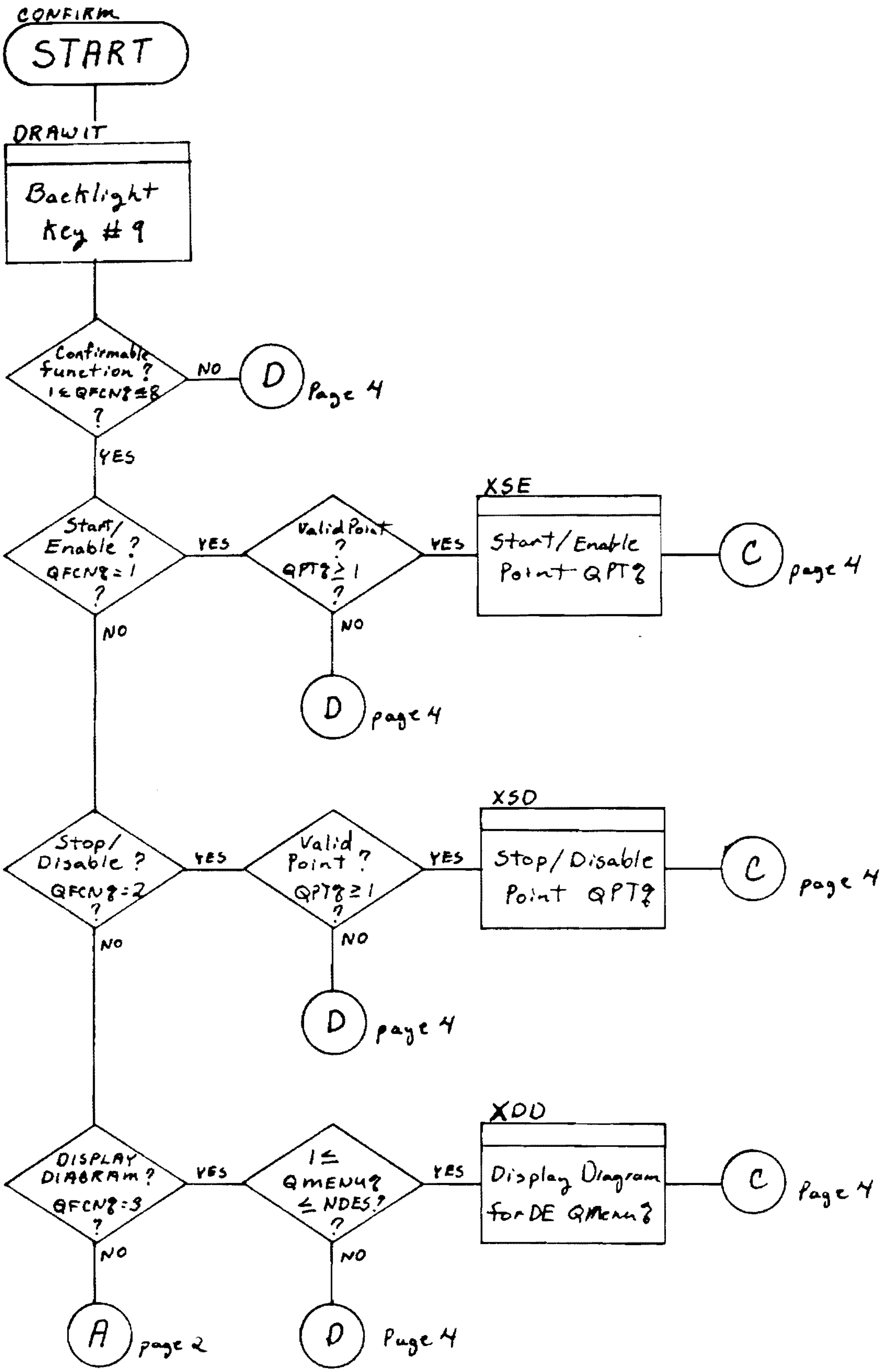
HARDWARE INTERACTION:

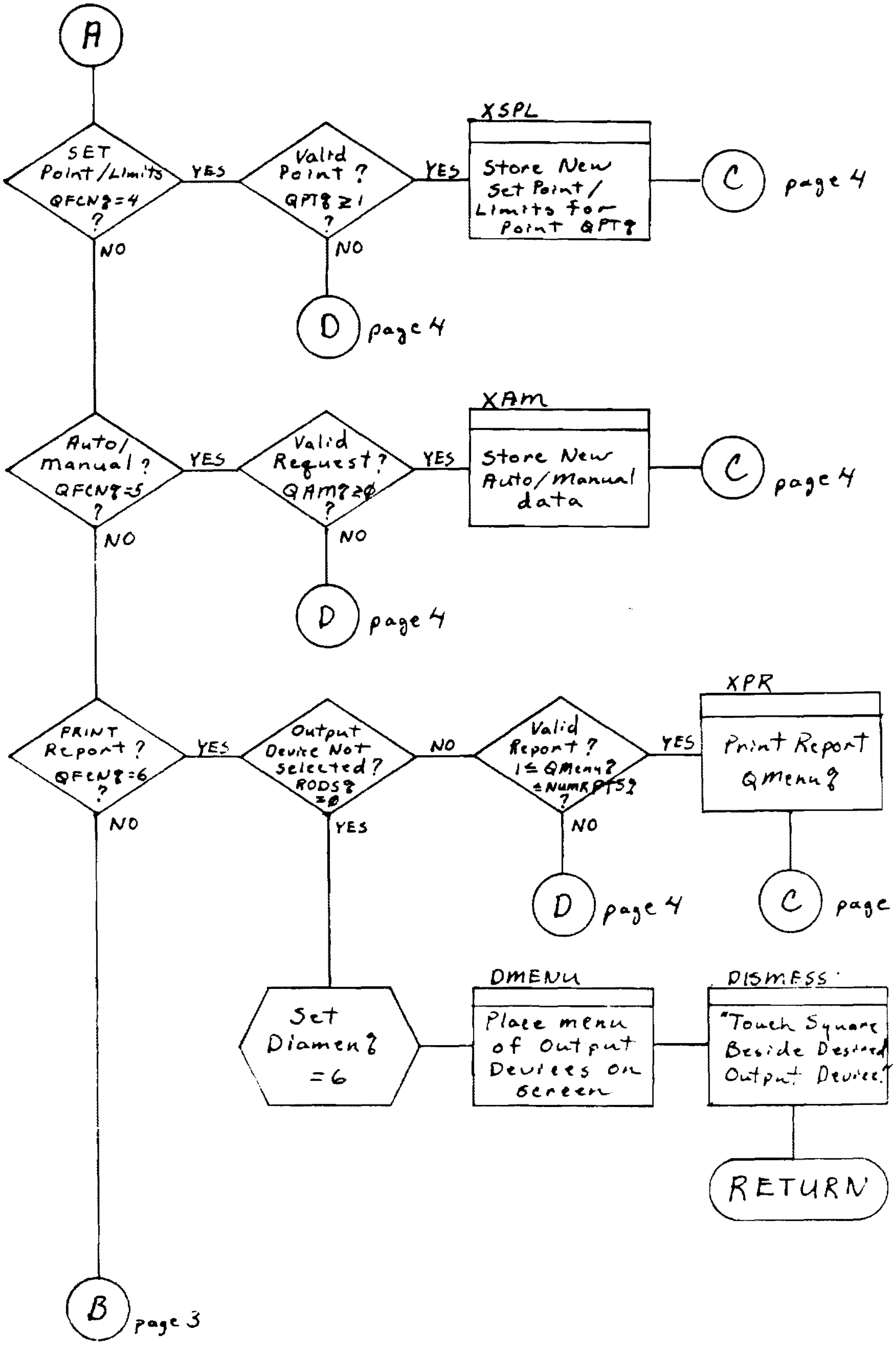
(none)

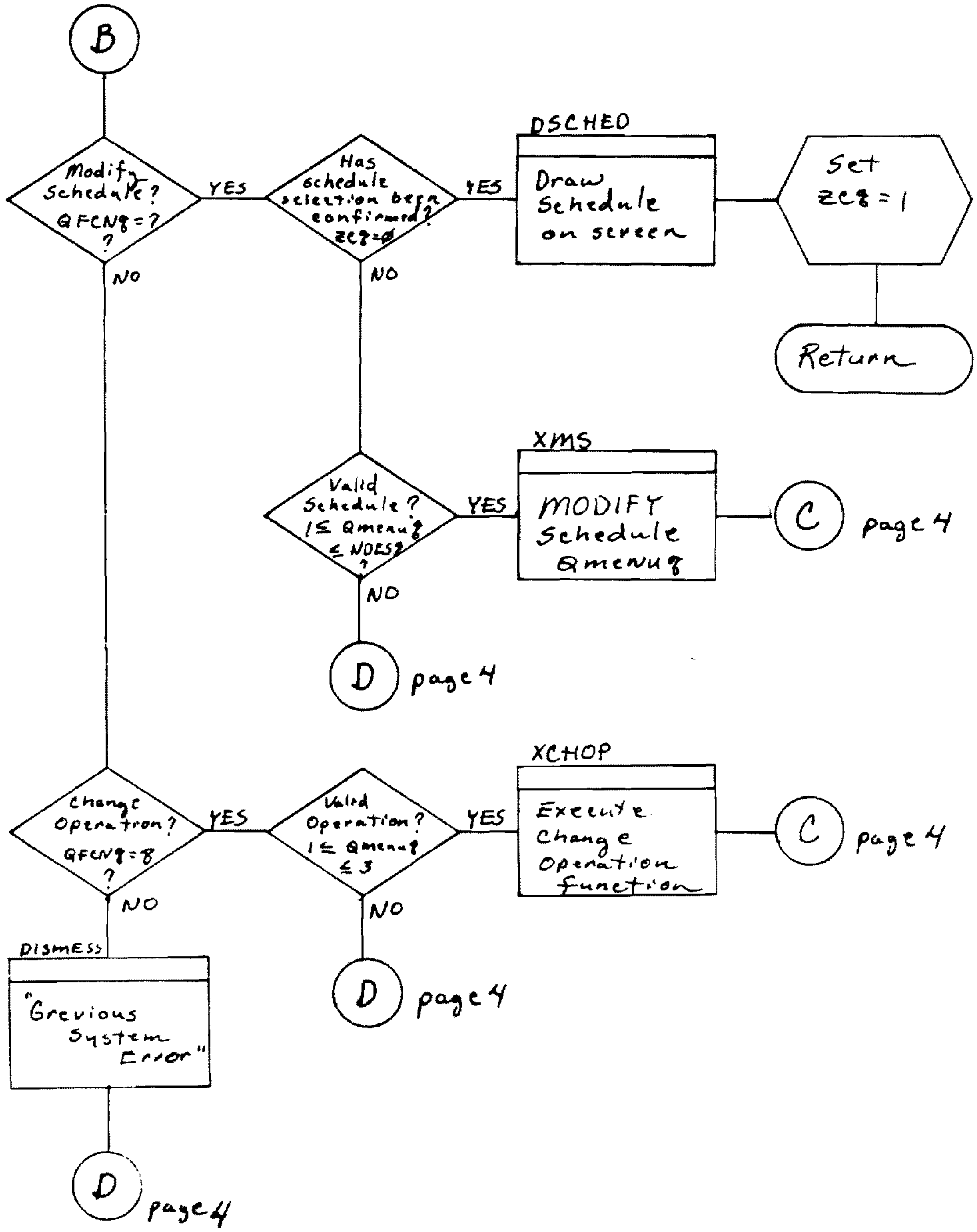
NAME: CONFIRM (continued)

DESIGN NOTES:

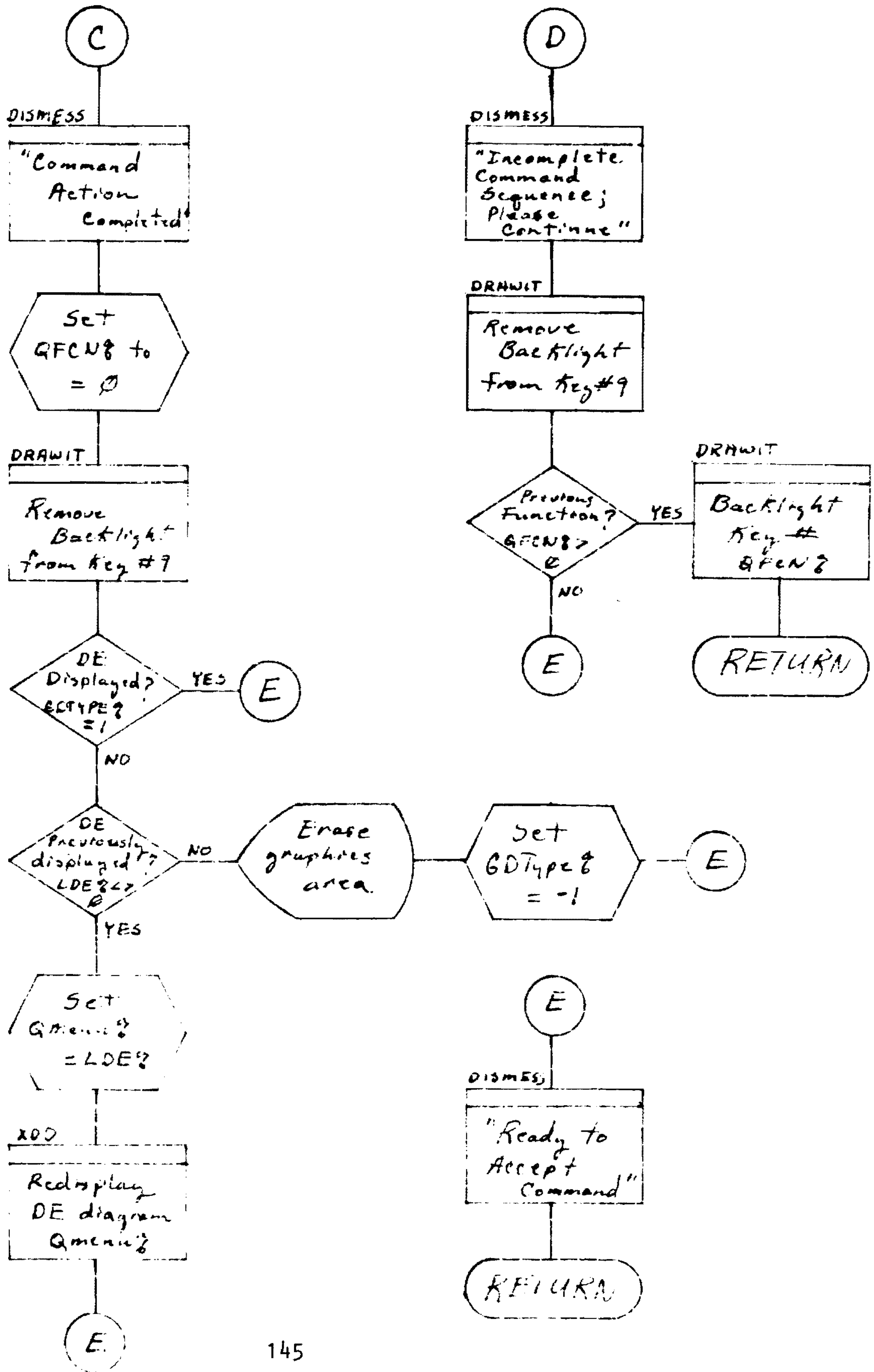
The CONFIRM module is usually only called once per function execution. However, there are two exceptions to this rule. First, the PRINT REPORT function requires the use of two levels of menu penetration. The CONFIRM module must be called once for each menu in order to process this function request. Second, the MODIFY SCHED function uses CONFIRM to verify schedule selection. When modifying the schedule, the CONFIRM module is called each time a schedule entry change is executed.


CONFIRM
START
DRAWIT
Backlight Key #9
Confirmable function? 1≤QFCN?≤8 ?
NO
D
Page 4
YES
Start/Enable? QFCN?=1 ?
YES
Valid Point? QPT?≥1 ?
YES
XSE
Start/Enable Point QPT?
C
page 4
NO
D
page 4
NO
Stop/Disable? QFCN?=2 ?
YES
Valid Point? QPT?≥1 ?
YES
XSD
Stop/Disable Point QPT?
C
page 4
NO
D
page 4
NO
DISPLAY DIAGRAM? QFCN?=3 ?
YES
1≤ QMENU? ≤NDES? ?
YES
XDD
Display Diagram for DE QMenu?
C
Page 4
NO
A
page 2
NO
D
Page 4


A
SET Point/Limits QFCN? = 4 ?
YES
NO
Valid Point ? QPT? ≥ 1 ?
YES
NO
XSPL
Store New Set Point/ Limits for Point QPT?
C page 4
D page 4
Auto/manual ? QFCN? = 5 ?
YES
NO
Valid Request ? QAM? ≥ 0 ?
YES
NO
XAM
Store New Auto/Manual data
C page 4
D page 4
PRINT Report ? QFCN? = 6 ?
YES
NO
Output Device Not Selected ? RODS? = 0
NO
YES
Valid Report ? 1 ≤ QMenu? ≤ NUMRPTS? ?
YES
NO
XPR
Print Report QMenu?
D page 4
C page 4
Set Diamen? = 6
DMENU
Place menu of Output Devices on Screen
DISMESS
"Touch Square Beside Desired Output Device"
RETURN
B page 3


B
Modify Schedule? QFCNQ=7 ?
YES
NO
Has schedule selection been confirmed? ZCQ=Ø
YES
NO
DSCHED
Draw Schedule on screen
Set ZCQ=1
Return
Valid Schedule? 1≤ QmenuQ ≤ NDESQ ?
YES
NO
XMS
MODIFY Schedule QmenuQ
C
page 4
D
page 4
Change Operation? QFCNQ=8 ?
YES
NO
Valid Operation? 1≤ QmenuQ ≤ 3
YES
NO
XCHOP
Execute Change Operation function
C
page 4
D
page 4
DISMESS
"Grevious System Error"
D
page 4

CONFIRM: PROCESS "CONFIRM ACTION" REQUEST Page 4 of 4



NAME: CANCEL

PURPOSE:

The CANCEL subroutine is used to process the CANCEL ACTION command request.

OPERATIONAL DESCRIPTION:

The CANCEL ACTION key is backlighted and the message "Command Action Cancelled" is displayed. Highlighting is removed, from the selected point, if any, and all selection variables are turned off. The function key backlight is removed the graphics area returns to its previous state, and program control returns to the calling routine.

CALLED BY:

FKEYIN

CALLS:

DISMESS

DRAWIT

XDD

PASSED ARGUMENTS:

GDTYPE%	- Graphics display type
LDE%	- Last Data Environment displayed
QPT%	- Selected point

NAME: CANCEL (continued)

RETURNED ARGUMENTS:

GDTYPE%	- Graphics display type
QAM%	- Selected mode
QFCN%	- Selected function
QMENU%	- Selected menu item
QPT%	- Selected point
QSCH%	- Selected schedule
ZC%	- Internal control variable

FILE INPUT/OUTPUT:

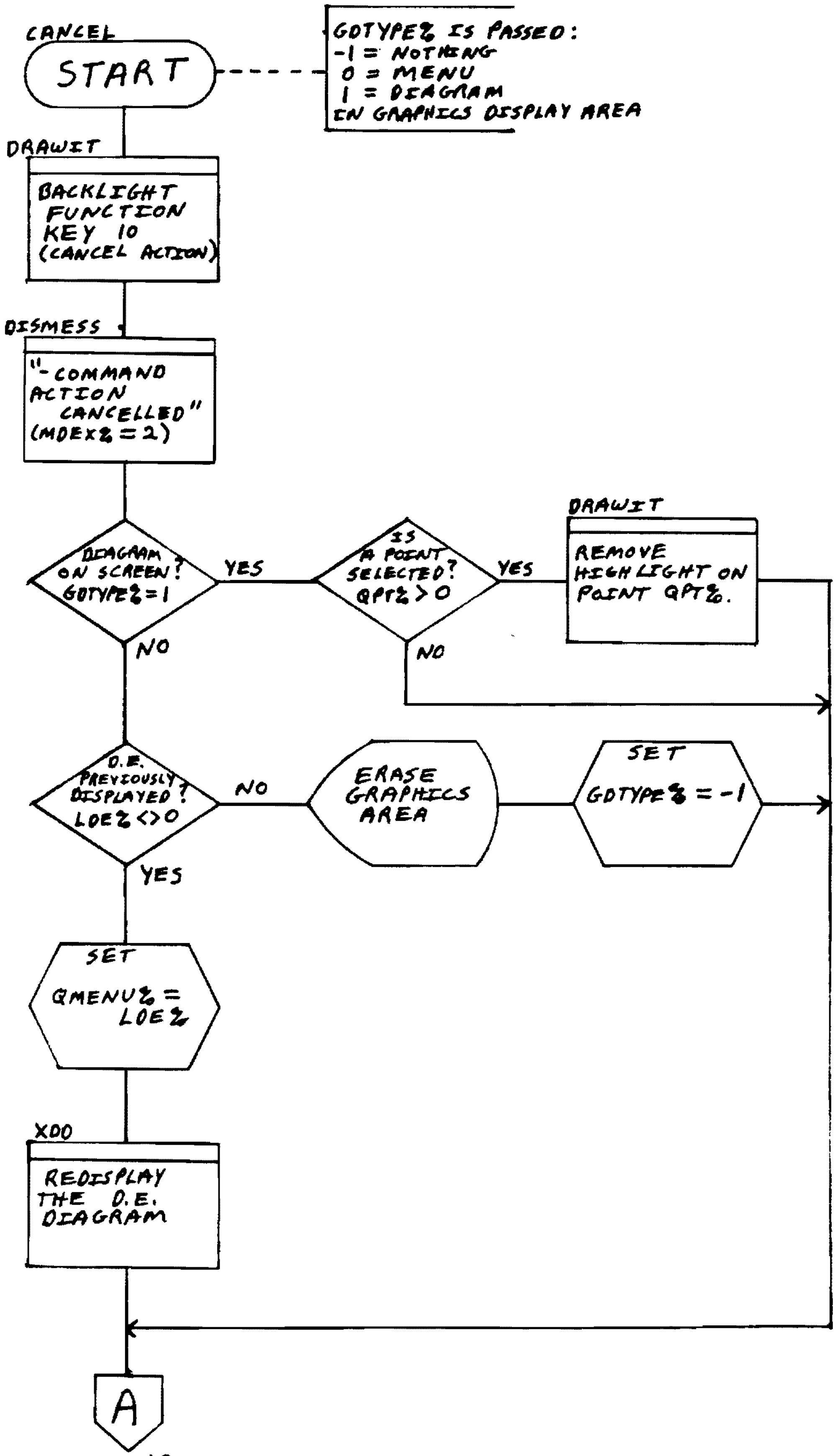
(none)

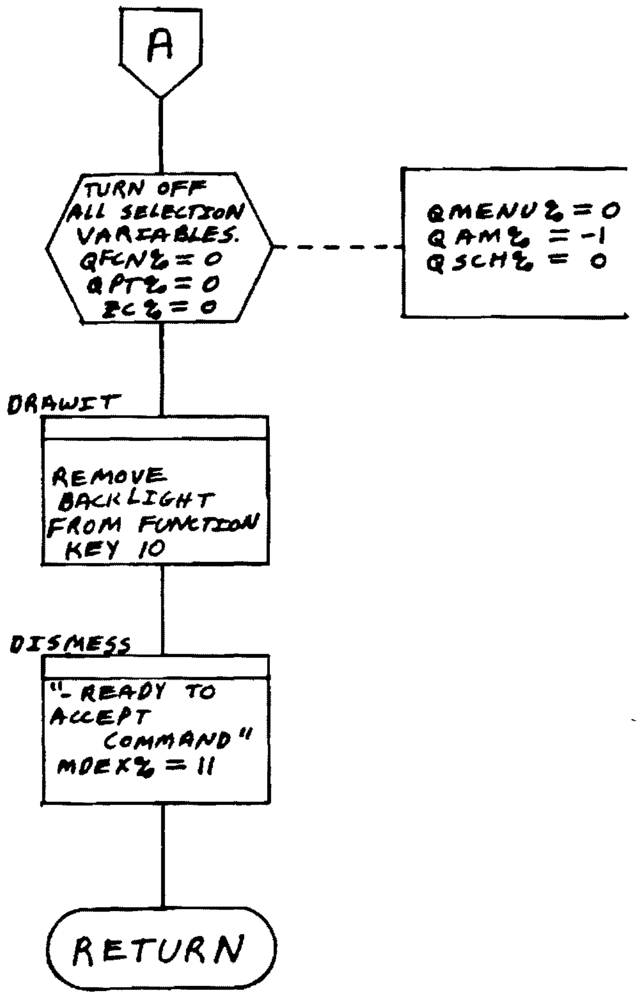
HARDWARE INTERACTION:

CRT DISPLAY - Manipulation of graphics area

DESIGN NOTES:

(none)


CANCEL
START
GOTYPE% IS PASSED:
-1 = NOTHING
0 = MENU
1 = DIAGRAM
IN GRAPHICS DISPLAY AREA
DRAWIT
BACKLIGHT FUNCTION KEY 10 (CANCEL ACTION)
DISMESS
"-COMMAND ACTION CANCELLED" (MDEX% = 2)
DIAGRAM ON SCREEN? GOTYPE% = 1
YES
IS A POINT SELECTED? QPT% > 0
YES
DRAWIT
REMOVE HIGHLIGHT ON POINT QPT%.
NO
NO
D.E. PREVIOUSLY DISPLAYED? LDE% <> 0
NO
ERASE GRAPHICS AREA
SET
GOTYPE% = -1
YES
SET
QMENU% = LDE%
XDO
REDISPLAY THE D.E. DIAGRAM
A


A
TURN OFF ALL SELECTION VARIABLES.
QFCN% = 0
QPT% = 0
ZC% = 0
QMENU% = 0
QAM% = -1
QSCH% = 0
DRAWIT
REMOVE BACKLIGHT FROM FUNCTION KEY 10
DISMESS
"- READY TO ACCEPT COMMAND"
MDEX% = 11
RETURN

NAME: MDIN

PURPOSE:

The MDIN subroutine is used to process the menu of diagrams selection.

OPERATIONAL DESCRIPTION:

The MDIN subroutine uses the coordinates of the touch panel input point to determine which menu item was selected. DRAWIT is then called to highlight the selected item and the menu selection variable is set.

CALLED BY:

MENUIN

CALLS:

DISMESS
DRAWIT

PASSED ARGUMENTS:

BOXLOC%	- Menu box location on screen, y-coordinate
DIAMEN%	- Number of diagram or menu currently displayed
NBOXES%	- Number of menu boxes for menus
YTUCH%	- Y-coordinate of touched point

RETURNED ARGUMENTS:

QMENU%	- Selected menu item

FILE INPUT/OUTPUT:

(none)

NAME: MDIN (continued)

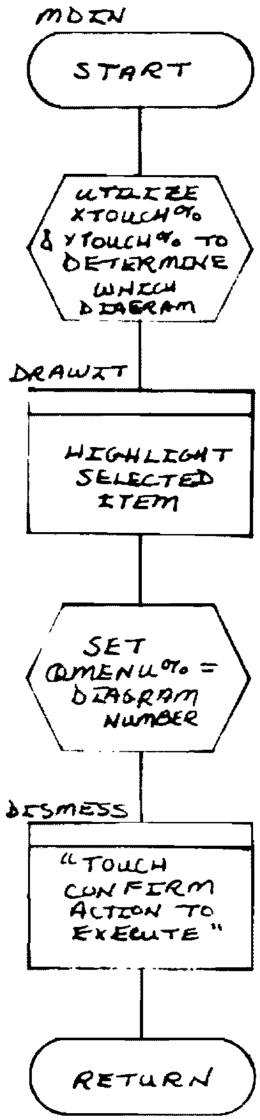
HARDWARE INTERACTION:

(none)

DESIGN NOTES:

(none)

MDIN: PROCESS MENU OF DIAGRAMS



NAME: MLIN

PURPOSE:

The MLIN subroutine is used to process the menu of SET POINTS/LIMITS selection request.

OPERATIONAL DESCRIPTION:

The coordinates of the touch panel input point are used to determine which item was selected. DRAWIT is called to highlight the selection, and keyboard input is enabled. When the new value is read from the keyboard it is checked to make sure that it is valid. An invalid value will result in an error message; otherwise, the value is stored in temporary storage. DRAWIT is then used to remove the highlight from the selection, and touch panel input is re-enabled.

CALLED BY:

MENUIN

CALLS:

DISMESS
DRAWIT
(KB ISR)

NAME: MLIN (continued)

PASSED ARGUMENTS:

BOXLOC%	-	Menu box location on screen, y-coordinate
DIAMEN%	-	Number of diagram or menu currently displayed
HTMP	-	High limit for analog point
LTMP	-	Low limit for analog point
NBOXES%	-	Number of menu boxes for menus
STMP	-	Set point for analog point
YTUCH%	-	Y-coordinate of touched point
ZIN$	-	Raw keyboard input

RETURNED ARGUMENTS:

HTMP	-	Temporary storage for analog high limit
LTMP	-	Temporary storage for analog low limit
QMENU%	-	Selected menu item
STMP	-	Temporary storage for analog set point

FILE INPUT/OUTPUT:

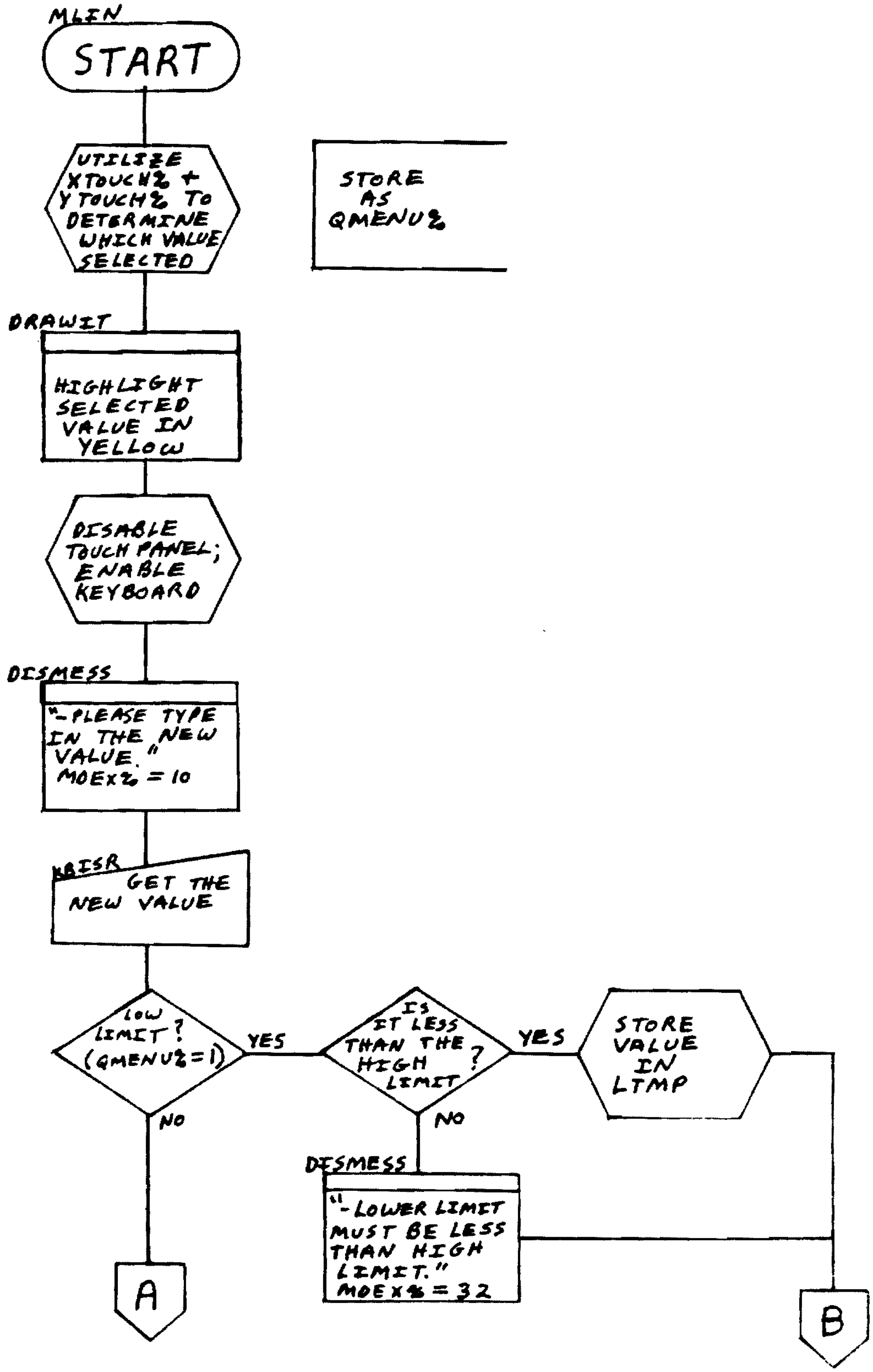
(none)

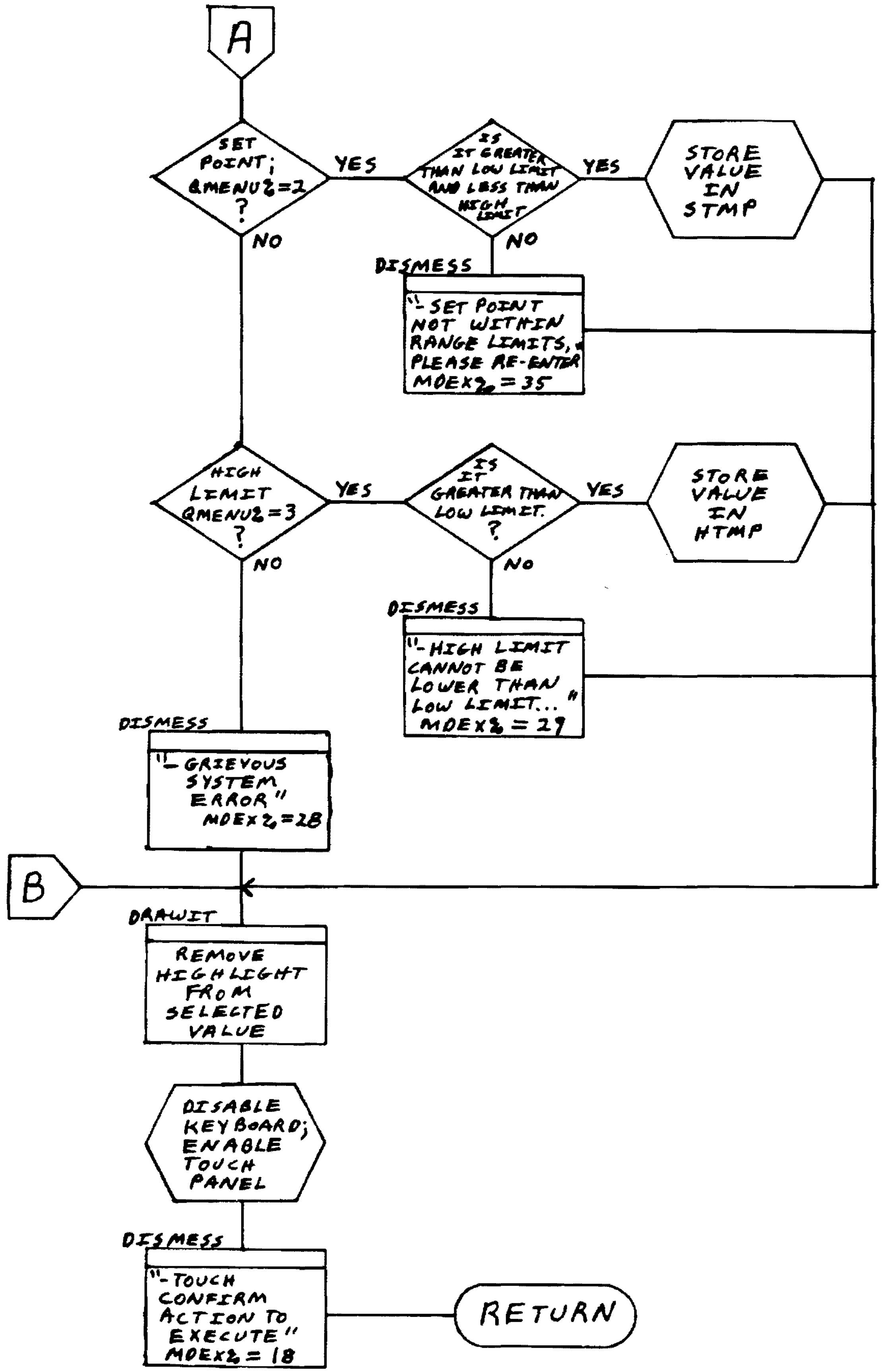
HARDWARE INTERACTION:

(none)

DESIGN NOTES:

(none)


MLIN
START
UTILIZE XTOUCH% + YTOUCH% TO DETERMINE WHICH VALUE SELECTED
STORE AS QMENU%
DRAWIT
HIGHLIGHT SELECTED VALUE IN YELLOW
DISABLE TOUCH PANEL; ENABLE KEYBOARD
DISMESS
"- PLEASE TYPE IN THE NEW VALUE." MOEX% = 10
KBISR
GET THE NEW VALUE
LOW LIMIT? (QMENU% = 1)
YES
NO
IS IT LESS THAN THE HIGH LIMIT?
YES
NO
STORE VALUE IN LTMP
DISMESS
"- LOWER LIMIT MUST BE LESS THAN HIGH LIMIT." MOEX% = 32
A
B


A
SET POINT; QMENU% = 2 ?
YES
NO
IS IT GREATER THAN LOW LIMIT AND LESS THAN HIGH LIMIT
YES
NO
STORE VALUE IN STMP
DISMESS
"- SET POINT NOT WITHIN RANGE LIMITS, PLEASE RE-ENTER"
MDEX% = 35
HIGH LIMIT QMENU% = 3 ?
YES
NO
IS IT GREATER THAN LOW LIMIT ?
YES
NO
STORE VALUE IN HTMP
DISMESS
"- HIGH LIMIT CANNOT BE LOWER THAN LOW LIMIT..."
MDEX% = 29
DISMESS
"- GRIEVOUS SYSTEM ERROR"
MDEX% = 28
B
DRAWIT
REMOVE HIGHLIGHT FROM SELECTED VALUE
DISABLE KEYBOARD; ENABLE TOUCH PANEL
DISMESS
"- TOUCH CONFIRM ACTION TO EXECUTE"
MDEX% = 18
RETURN

NAME: MOIN

PURPOSE:

The MOIN subroutine is used to process the menu of operations selection.

OPERATIONAL DESCRIPTION:

The coordinates of the touch panel input point are used to determine which menu item has been selected. DRAWIT is called to highlight the selection, and the selection control variable is set.

CALLED BY:

MENUIN

CALLS:

DISMESS

DRAWIT

PASSED ARGUMENTS:

BOXLOC%	- Menu box location on screen, y coordinate
DIAMEN%	- Number of diagram or menu currently displayed
NBOXES%	- Number of menu boxes for menus
YTUCH%	- Y-coordinate of touched point

RETURNED ARGUMENTS:

QMENU%	- Selected menu item.

NAME: MOIN (continued)

FILE INPUT/OUTPUT:

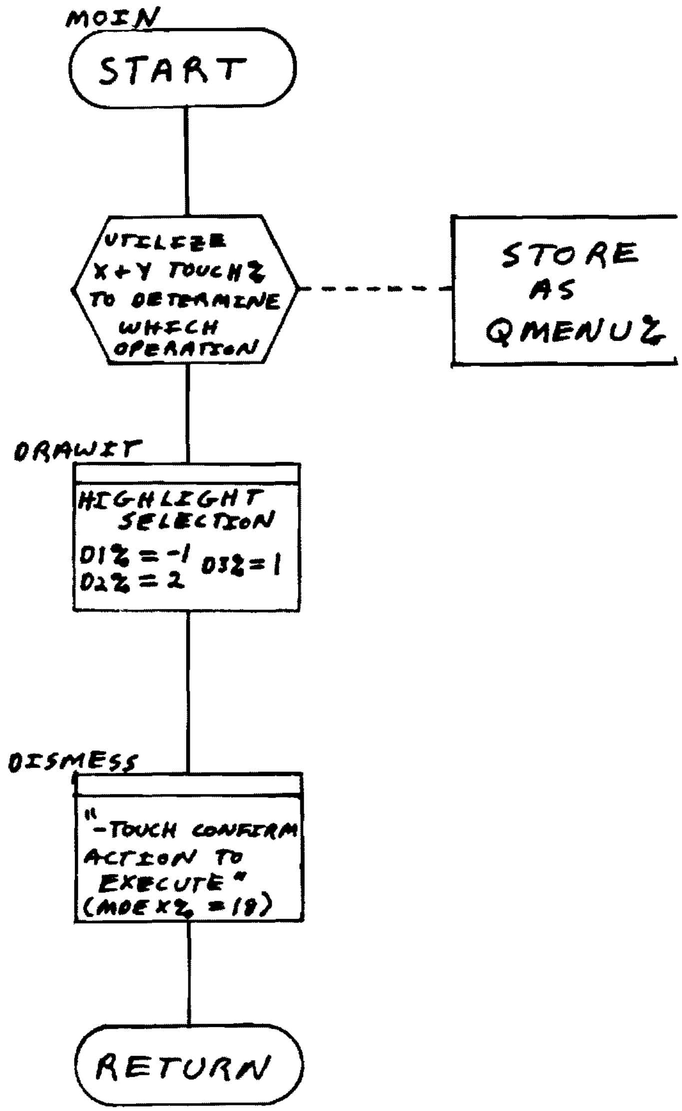
(none)

HARDWARE INTERACTION:

(none)

DESIGN NOTES:

(none)


MOIN
START
UTILIZE X+Y TOUCH% TO DETERMINE WHICH OPERATION
STORE AS QMENU%
DRAWIT
HIGHLIGHT SELECTION
D1% = -1
D2% = 2
D3% = 1
DISMESS
"-TOUCH CONFIRM ACTION TO EXECUTE"
(MDEX% = 19)
RETURN

NAME: MRIN

PURPOSE:

The MRIN subroutine is used to process the menu of reports selection.

OPERATIONAL DESCRIPTION:

The coordinates of the touch panel input point are used to determine which menu item has been selected. DRAWIT is called to highlight the selection, and the selection control variable is set.

CALLED BY:

MENUIN

CALLS:

DISMESS

DRAWIT

PASSED ARGUMENTS:

BOXLOC%	- Menu box location on screen, y-coordinate
DIAMEN%	- Number of diagram or menu currently displayed
NBOXES%	- Number of menu boxes for menus
YTUCH%	- Y-coordinate of touched point

RETURNED ARGUMENTS:

QMENU%	- Selected menu item.

NAME: MRIN (continued)

FILE INPUT/OUTPUT:

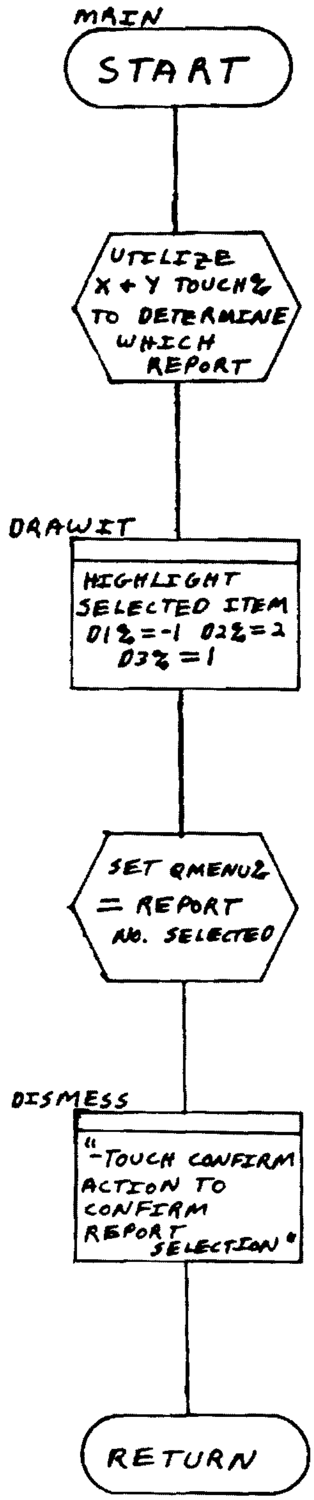
(none)

HARDWARE INTERACTION:

(none)

DESIGN NOTES:

(none)


MAIN
START
UTILIZE X + Y TOUCH% TO DETERMINE WHICH REPORT
DRAWIT
HIGHLIGHT SELECTED ITEM D1%=-1 D2%=2 D3%=1
SET QMENU% = REPORT NO. SELECTED
DISMESS
"-TOUCH CONFIRM ACTION TO CONFIRM REPORT SELECTION"
RETURN

NAME: MSIN

PURPOSE:

The MSIN subroutine is used to process the menu of schedules selection.

OPERATIONAL DESCRIPTION:

The coordinates of the touch panel input point are used to determine which menu item was selected. DRAWIT is used to highlight the selection, and the selection control variable is set.

CALLED BY:

MENUIN

CALLS:

DISMESS

DRAWIT

PASSED ARGUMENTS:

BOXLOC%	- Menu box location on screen, y-coordinate
DIAMEN%	- Number of diagram or menu currently displayed
NBOXES%	- Number of menu boxes for menus
Y-TUCH%	- Y-coordinate of touched point

NAME: MSIN (continued)

RETURNED ARGUMENTS:

QMENU% - Selected menu item

QSCH% - Selected schedule

FILE INPUT/OUTPUT:

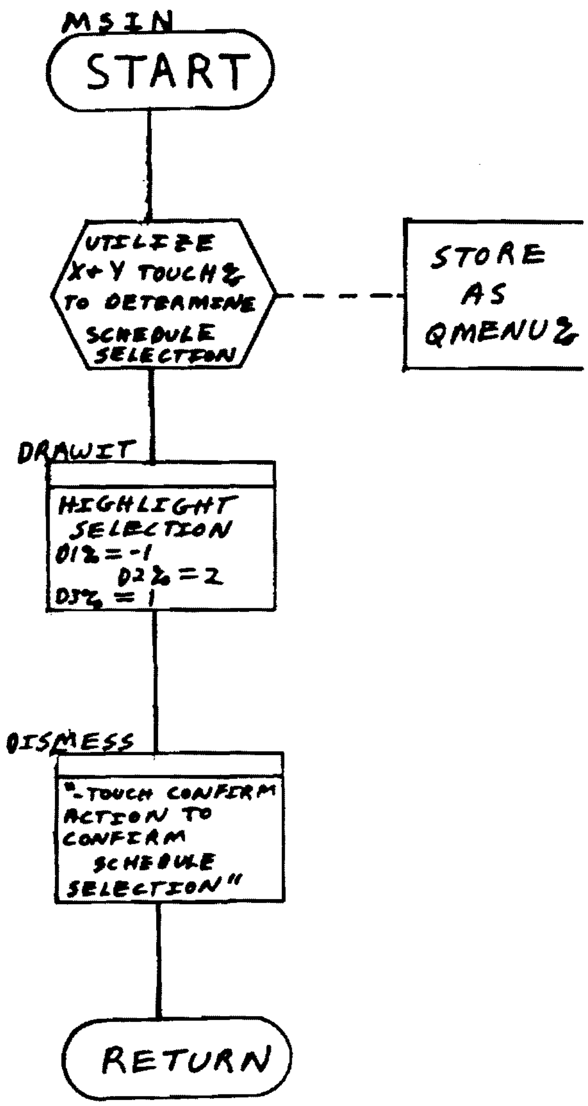
(none)

HARDWARE INTERACTION:

(none)

DESIGN NOTES:

(none)


MSIN
START
UTILIZE X+Y TOUCH% TO DETERMINE SCHEDULE SELECTION
STORE AS QMENU%
DRAWIT
HIGHLIGHT SELECTION
D1% = -1
D2% = 2
D3% = 1
DISMESS
"-TOUCH CONFIRM ACTION TO CONFIRM SCHEDULE SELECTION"
RETURN

NAME: MSCHED

PURPOSE:

The MSCHED subroutine is used to process a MODIFY SCHED request.

OPERATIONAL DESCRIPTION:

The coordinates of the touch panel input point are used to determine which scehdule entry is to be modified. MSCHED highlights the selection, and keyboard input is enabled. When the new schedule value is read from the keyboard, it is checked for validity. An invalid value will result in an error message; otherwise the value is stored in temporary storage. The screen is then updated to reflect the new value, and touch panel input is re-enabled.

CALLED BY:

MENUIN

CALLS:

DISMESS

(KB ISR)

PASSED ARGUMENTS:

NSE%	- Number of schedule entries
SCHTMP$	- Temporary schedule storage
VPS%	- Schedule entry coordinates
XTUCH%	- X-coordinate of touch point
YTUCH%	- Y-coordinate of touch point
ZIN$	- Raw keyboard input

NAME: MSCHED (continued)

RETURNED ARGUMENTS:

SCHTMP$ - Modified schedule

FILE INPUT/OUT:

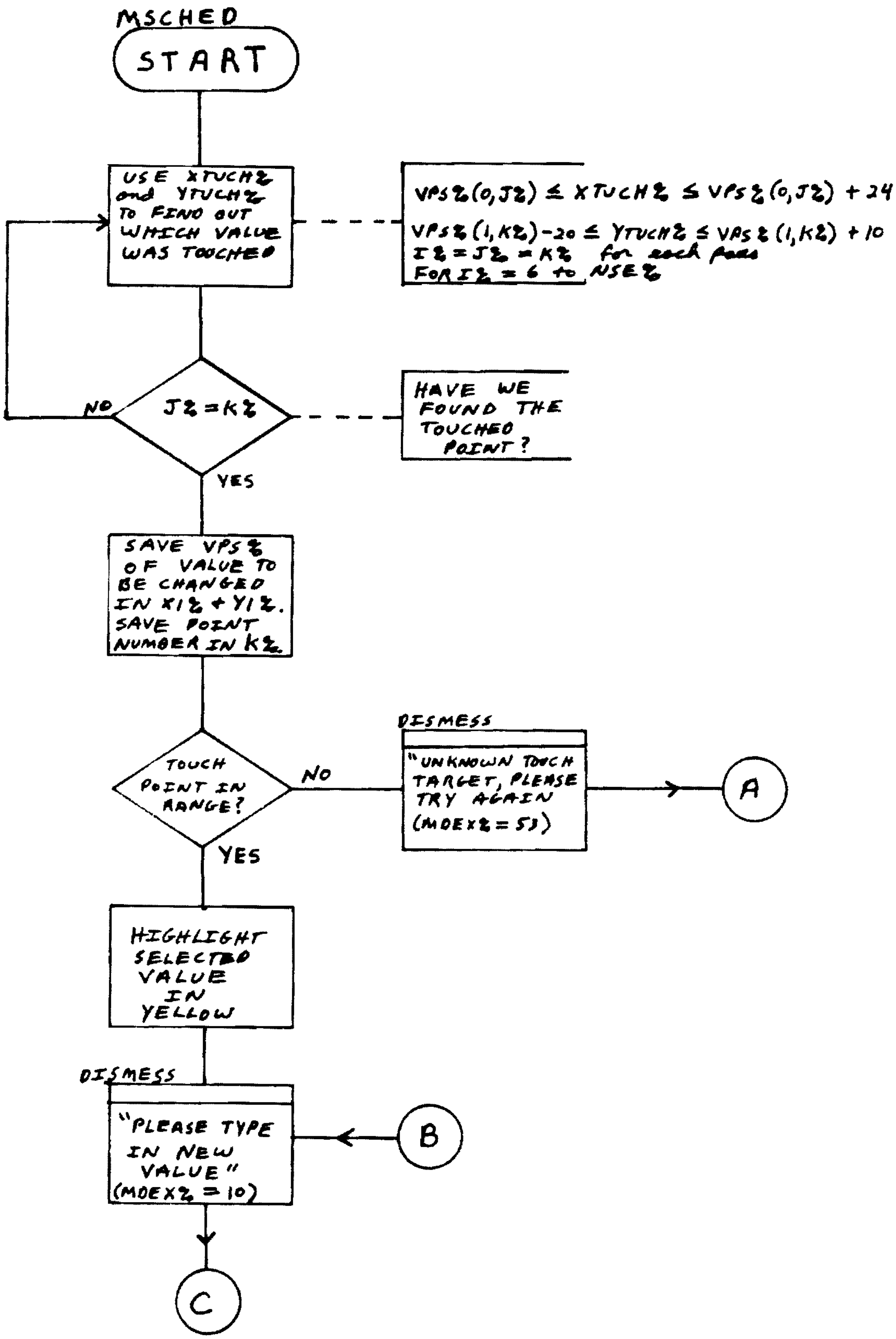
(none)

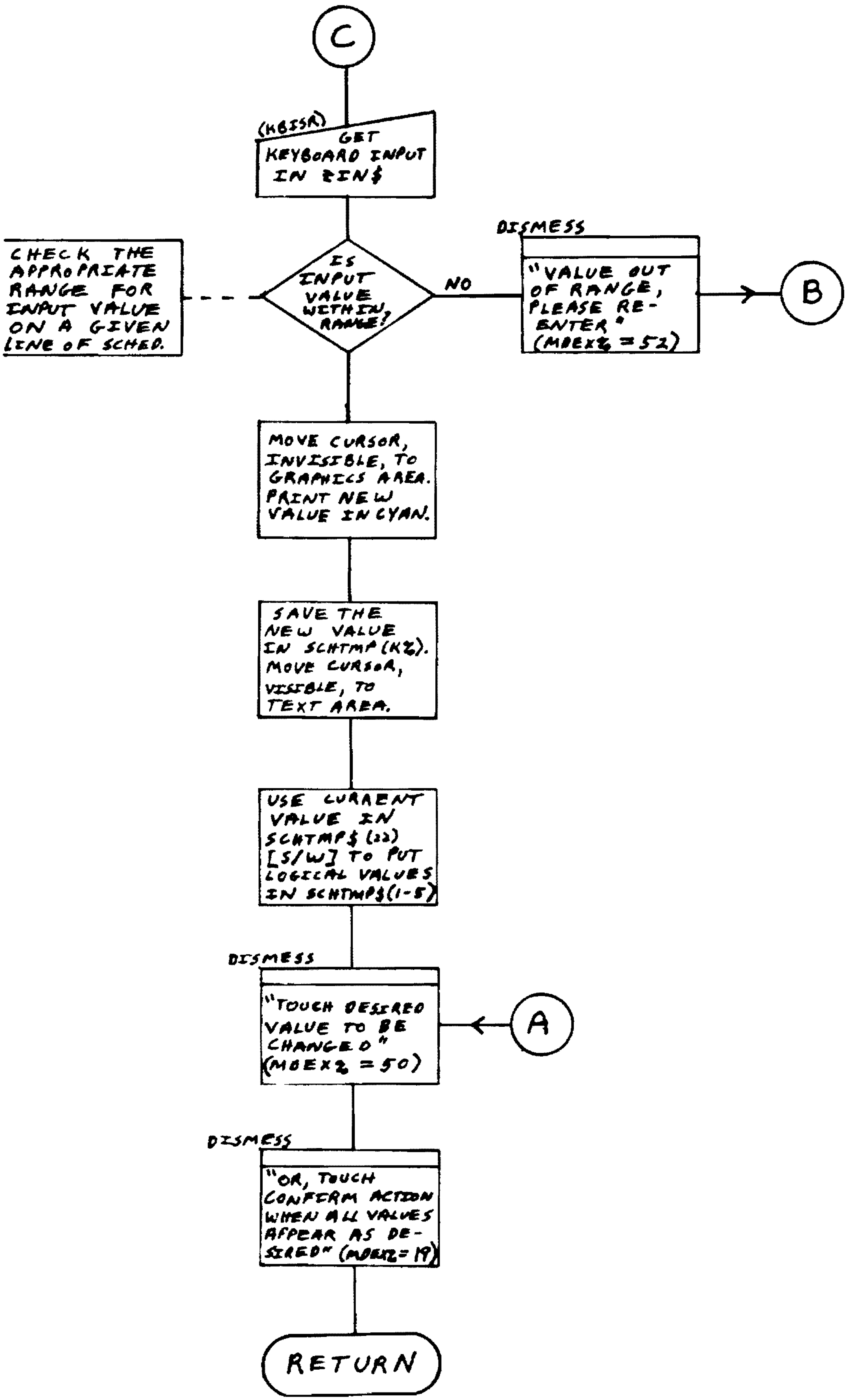
HARDWARE INTERACTION:

CRT Screen - Modify displayed values

DESIGN NOTES:

(none)


MSCHED
START
USE XTUCH% and YTUCH% TO FIND OUT WHICH VALUE WAS TOUCHED
VPS%(0,J%) ≤ XTUCH% ≤ VPS%(0,J%) + 24
VPS%(1,K%)-20 ≤ YTUCH% ≤ VPS%(1,K%) + 10
I% = J% = K% for each pass
FOR I% = 6 to NSE%
J% = K%
NO
YES
HAVE WE FOUND THE TOUCHED POINT?
SAVE VPS% OF VALUE TO BE CHANGED IN X1% + Y1%. SAVE POINT NUMBER IN K%.
TOUCH POINT IN RANGE?
NO
YES
DISMESS
"UNKNOWN TOUCH TARGET, PLEASE TRY AGAIN (MDEX% = 53)
A
HIGHLIGHT SELECTED VALUE IN YELLOW
DISMESS
"PLEASE TYPE IN NEW VALUE" (MDEX% = 10)
B
C


C
(KBISR)
GET KEYBOARD INPUT IN ZIN$
CHECK THE APPROPRIATE RANGE FOR INPUT VALUE ON A GIVEN LINE OF SCHED.
IS INPUT VALUE WITHIN RANGE?
NO
DISMESS
"VALUE OUT OF RANGE, PLEASE RE-ENTER" (MDEX% = 52)
B
MOVE CURSOR, INVISIBLE, TO GRAPHICS AREA. PRINT NEW VALUE IN CYAN.
SAVE THE NEW VALUE IN SCHTMP(K%). MOVE CURSOR, VISIBLE, TO TEXT AREA.
USE CURRENT VALUE IN SCHTMP$ (22) [S/W] TO PUT LOGICAL VALUES IN SCHTMP$(1-5)
DISMESS
"TOUCH DESIRED VALUE TO BE CHANGED" (MDEX% = 50)
A
DISMESS
"OR, TOUCH CONFIRM ACTION WHEN ALL VALUES APPEAR AS DESIRED" (MDEX% = 19)
RETURN

NAME: REPOUT

PURPOSE:

The REPOUT subroutine is used to process the menu of report output devices selection.

OPERATIONAL DESCRIPTION:

The coordinates of the touch panel input point are used to determine which menu item has been selected. DRAWIT is called to highlight the selection, and the selection control variable is set.

CALLED BY:

MENUIN

CALLS:

DISMESS

DRAWIT

PASSED ARGUMENTS:

BOXLOC%	- Menu box location on screen, y-coordinate
DIAMEN%	- Number of diagram or menu currently displayed
NBOXES%	- Number of menu boxes for menus
YTUCH%	- Y-coordinate of touch point

RETURNED ARGUMENTS:

RODS%	- Selected menu item

NAME: REPOUT (continued)

FILE INPUT/OUTPUT:

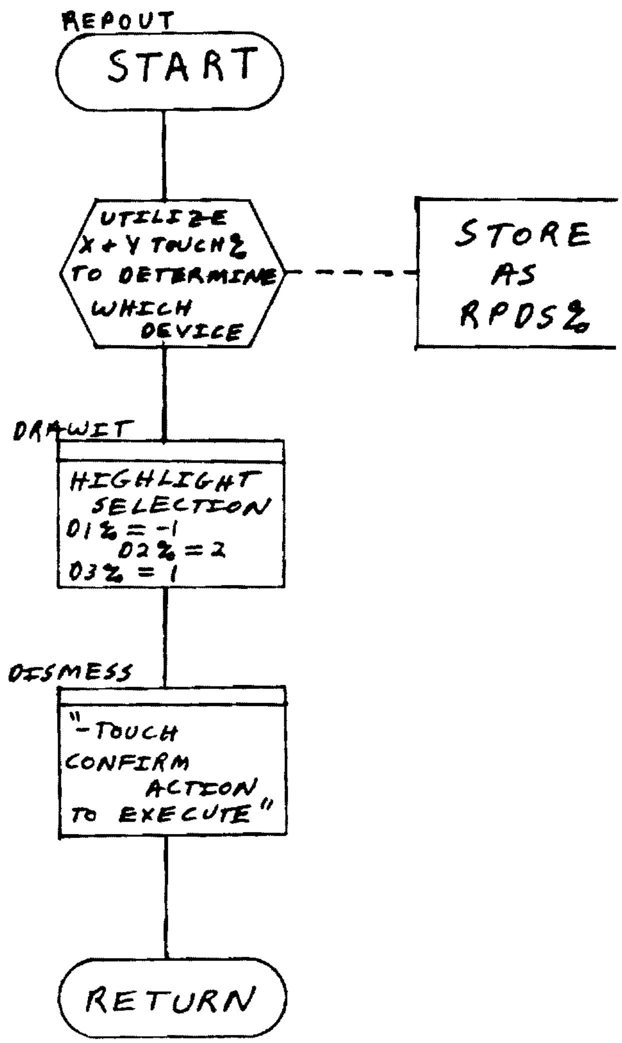
(none)

HARDWARE INTERACTION:

(none)

DESIGN NOTES:

This module is used in conjunction with a level 2 menu penetration. See the Design Notes on the CONFIRM module for further details.


REPOUT
START
UTILIZE X + Y TOUCH% TO DETERMINE WHICH DEVICE
STORE AS RPDS%
DRAWIT
HIGHLIGHT SELECTION
D1% = -1
D2% = 2
D3% = 1
DISMESS
"-TOUCH CONFIRM ACTION TO EXECUTE"
RETURN

NAME: XSE

PURPOSE:

The XSE subroutine is used to execute the START/ENABLE function.

OPERATIONAL DESCRIPTION:

The XSE subroutine sets the status indicator for the selected point to the started/enabled condition. In addition, it enables all devices tied to and below the selected point in the DE process heirarchy. If the selected point is a start/stopable point then all start/stopable devices tied to and below the selected point in the DE process hierarchy will also be started. It also readjusts the symbol coloration for the affected devices if they are currently displayed on the screen. Messages are given for each device which is affected. If the selected point is the current alarm point, ADISABLE is called to disable the alarm.

CALLED BY:

CONFIRM

CALLS:

DRAWIT

ADISABLE

NAME: XSE (continued)

PASSED ARGUMENTS:

APT%	- Current alarm point
DIAMEN%	- Number of diagram or menu currently displayed
LPNT%	- Number of diagram on which point is located
NRTPTS%	- Number of real-time points
PHIER%	- Pointer to next higher device in hierarchy
PNAME$	- Textual name of point
PTYPE%	- Type of point
QPT%	- Selected point
SSED%	- Start/Stop - Enable/Disable device discriminator

RETURNED ARGUMENTS:

CDBVL	- Current database value for Ith point
DSTAT%	- Device status
QPT%	- Selected point

FILE INPUT/OUTPUT:

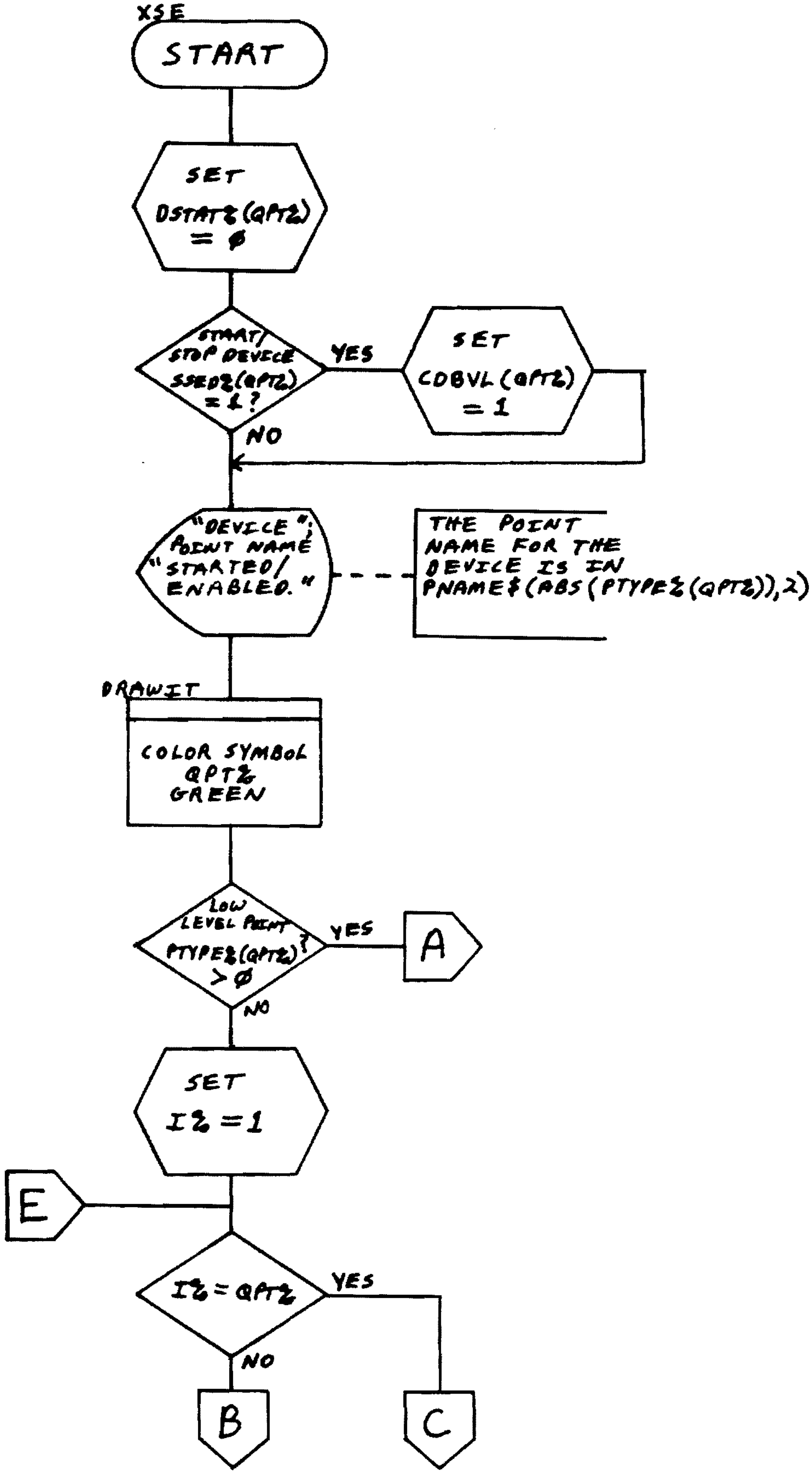
(none)

HARDWARE INTERACTION:

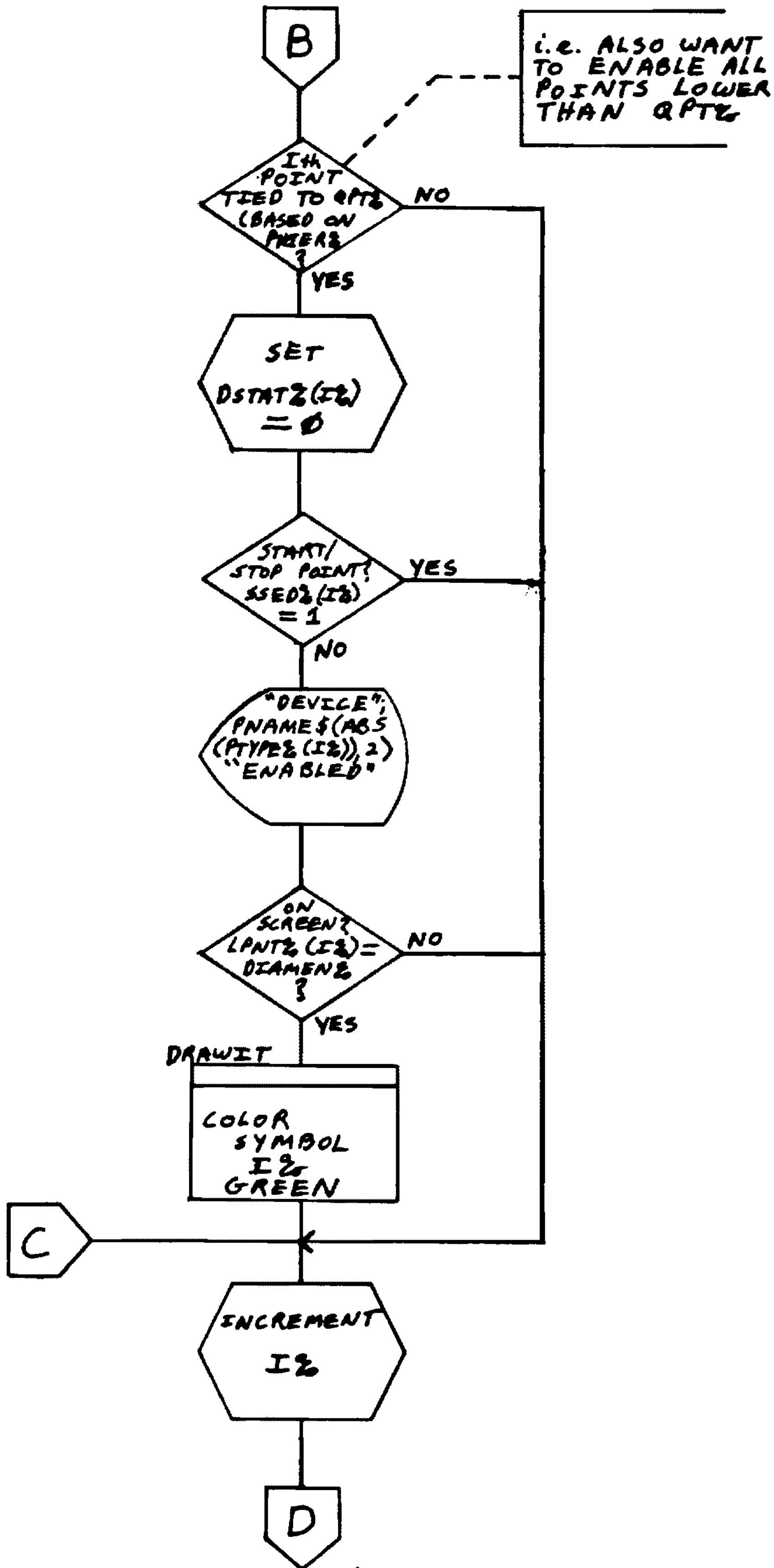
CRT Screen - prints messages

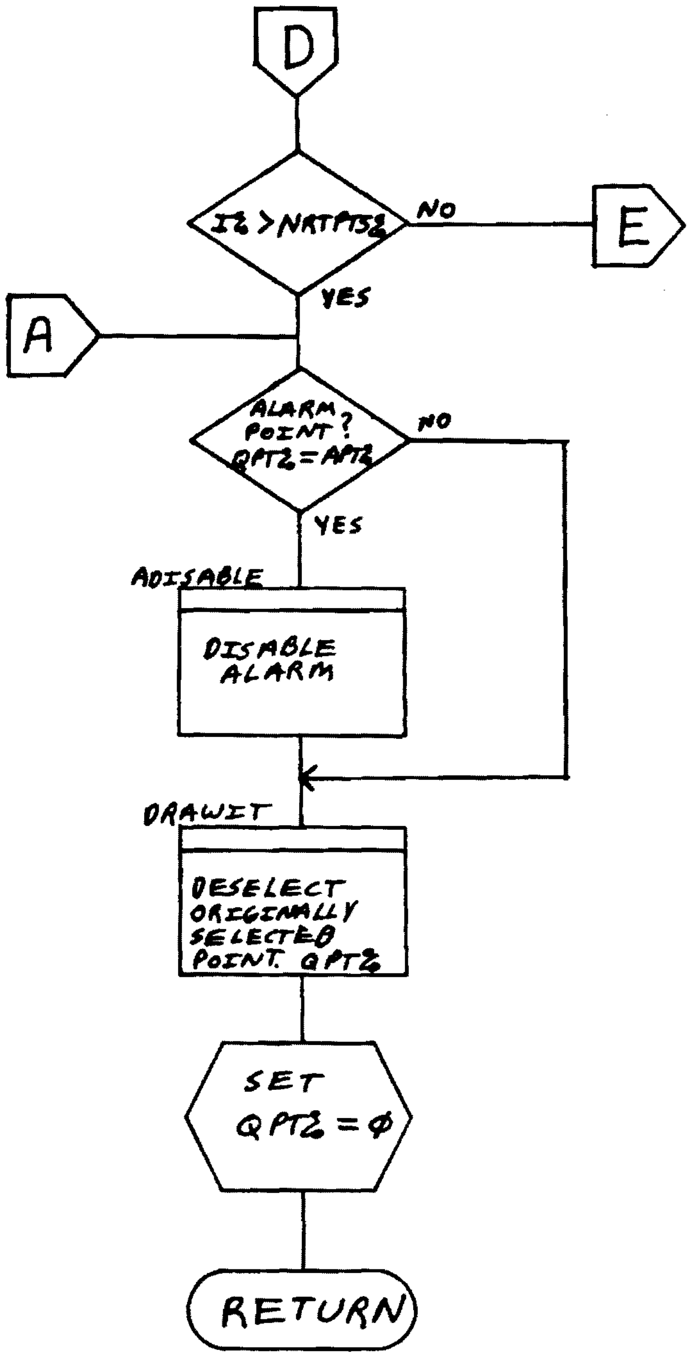
DESIGN NOTES:

(none)



XSE
START
SET
DSTAT%(QPT%)
= Ø
START/
STOP DEVICE
SSED%(QPT%)
= 1?
YES
SET
CDBVL(QPT%)
= 1
NO
"DEVICE";
POINT NAME
"STARTED/
ENABLED."
THE POINT
NAME FOR THE
DEVICE IS IN
PNAME$(ABS(PTYPE%(QPT%)),2)
DRAWIT
COLOR SYMBOL
QPT%
GREEN
LOW
LEVEL POINT
PTYPE%(QPT%)?
> Ø
YES
A
NO
SET
I% = 1
E
I% = QPT%
YES
NO
B
C


B
i.e. ALSO WANT TO ENABLE ALL POINTS LOWER THAN QPT%
Ith POINT TIED TO QPT% (BASED ON PHIER%) ?
NO
YES
SET DSTAT%(I%) = Ø
START/STOP POINT? SSED%(I%) = 1
YES
NO
"DEVICE"; PNAME$(ABS(PTYPE%(I%)),2) "ENABLED"
ON SCREEN? LPNT%(I%) = DIAMEN% ?
NO
YES
DRAWIT
COLOR SYMBOL I% GREEN
C
INCREMENT I%
D


D
I% > NRTPTS%
NO
E
YES
A
ALARM POINT? QPT% = APT%
NO
YES
ADISABLE
DISABLE ALARM
DRAWIT
DESELECT ORIGINALLY SELECTED POINT QPT%
SET QPT% = Φ
RETURN

NAME: XSD

PURPOSE:

The XSD subroutine is used to execute the STOP/DISABLE function.

OPERATIONAL DESCRIPTION:

The XSD subroutine sets the status indicator for the selected point to the stopped/disabled condition. In addition all non start/stopable points which are tied to and below it in the DE process hierarchy are also disabled. If the original request was a stop, all stopable devices which are tied to and below the point are also stopped. It also readjusts the symbol coloration for the affected devices if they are currently displayed on the screen. Messages are given for each device which is affected. If the selected device is the current alarm point, then ADISABLE is called to disable the alarm.

CALLED BY:

CONFIRM

CALLS:

DRAWIT

ADISABLE

NAME: XSD

PASSED ARGUMENTS:

APT%	- Current alarm point
DIAMEN%	- Number of diagram or menu currently displayed
LPNT%	- Number of diagram on which point is located
NRTPTS%	- Number of real-time points
PHIER%	- Pointer to next higher device in hierarchy
PNAME$	- Textual name of point
PTYPE%	- Type of point
QPT%	- Selected point
SSED%	- Start/Stop - Enable/Disable Device discrimintator

RETURNED ARGUMENTS:

CDBVL	- Current database value for Ith point
DSTAT%	- Device status
QPT%	- Selected point

FILE INPUT/OUTPUT:

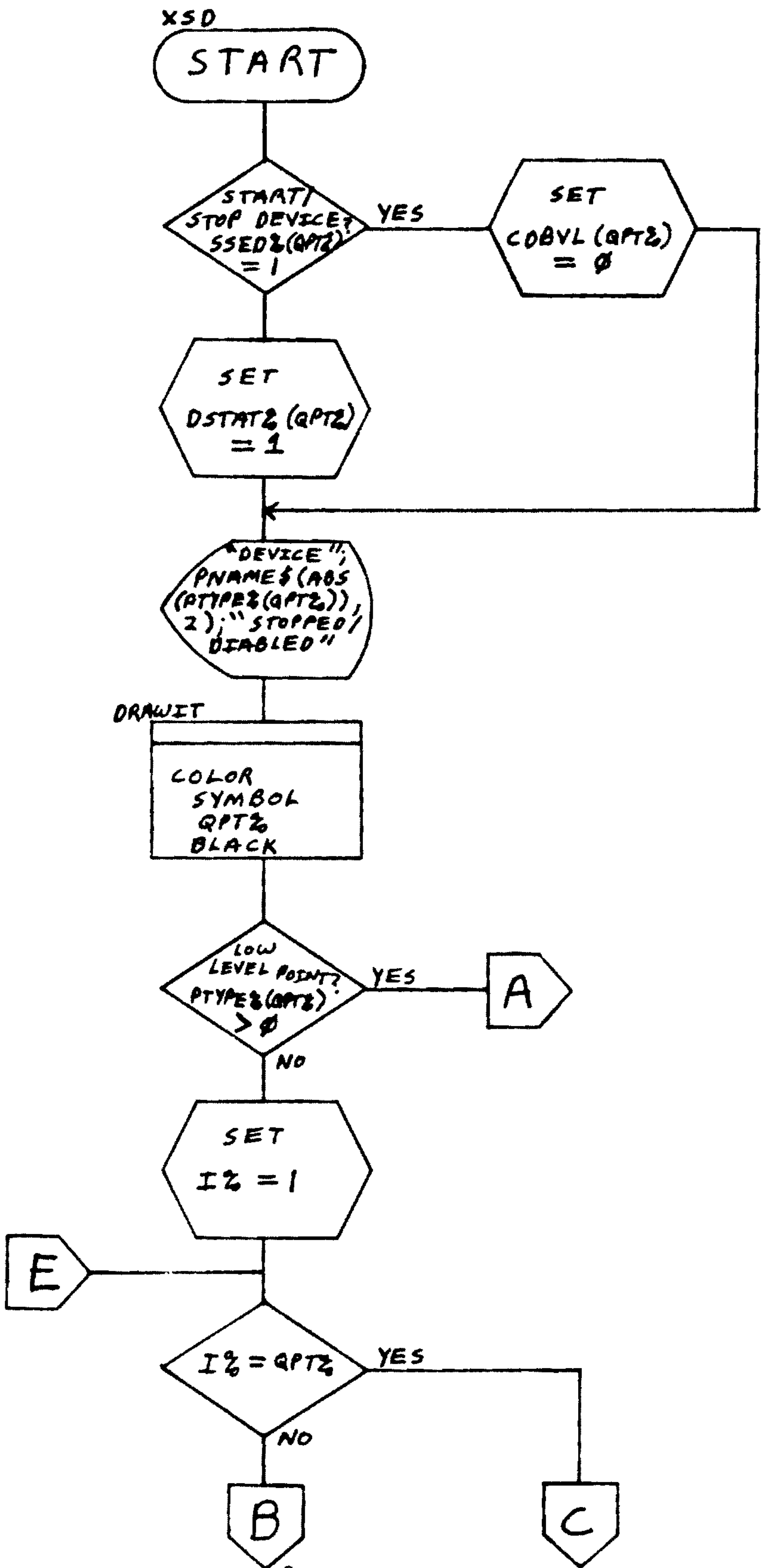
(none)

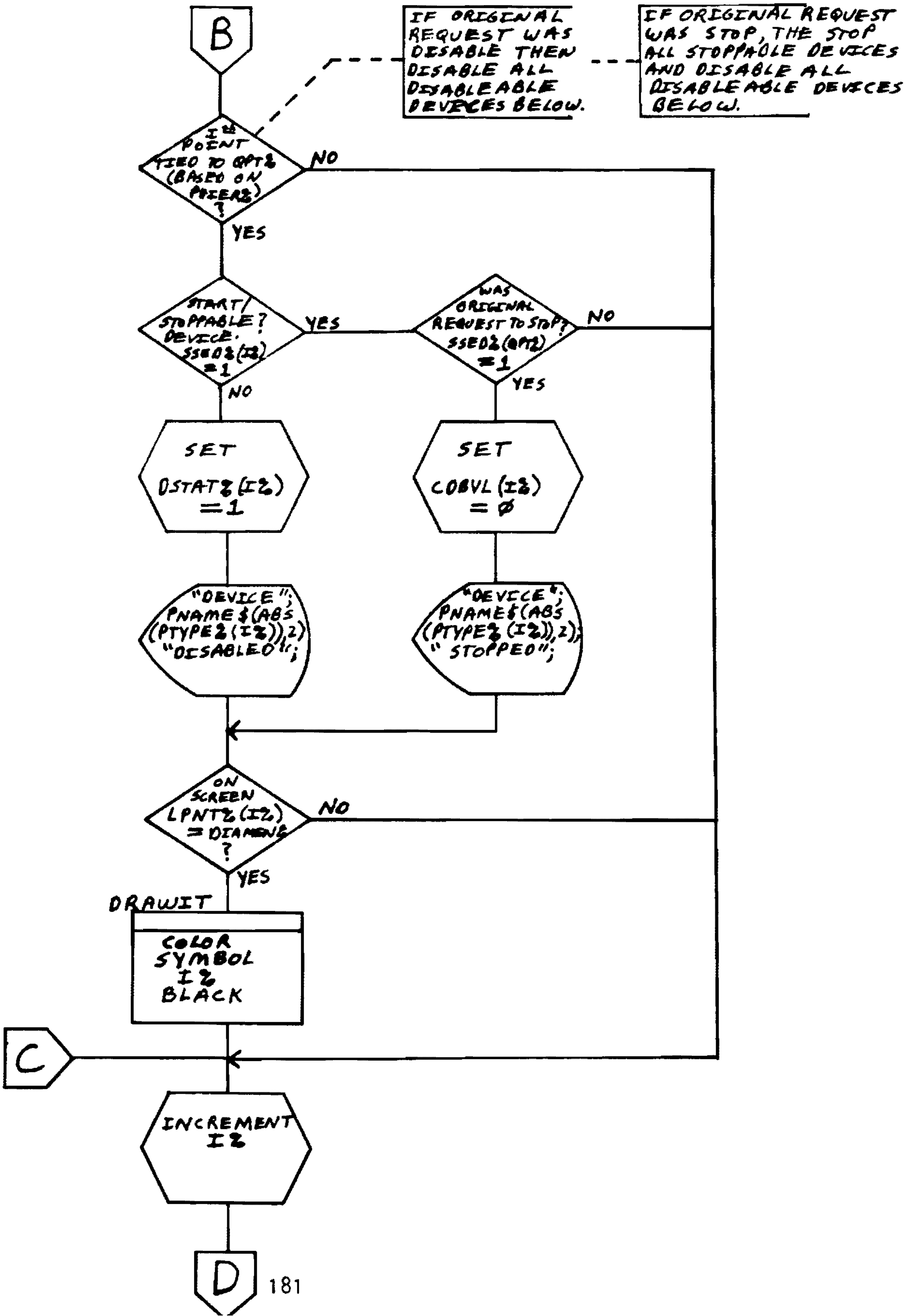
HARDWARE INTERACTION:

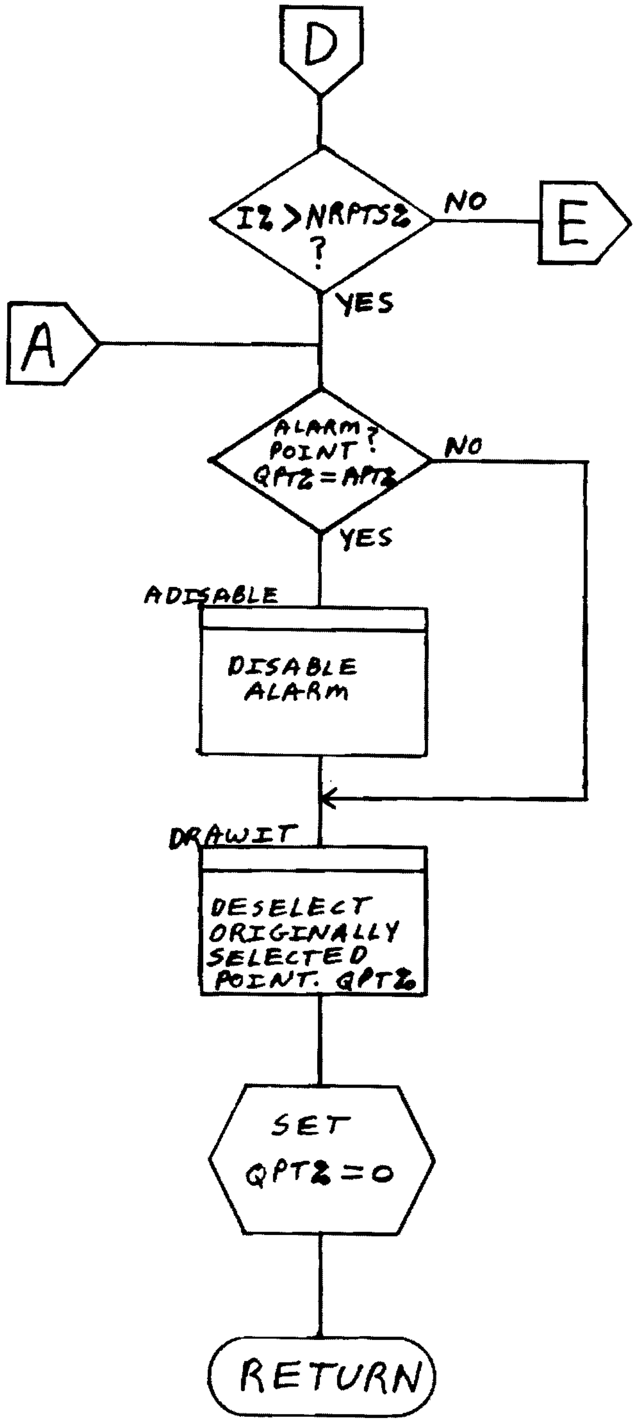
CRT Screen	- Prints messages to text window

DESIGN NOTES:

(none)


XSD
START
START/STOP DEVICE? SSED%(QPT%) = 1
YES
SET COBVL (QPT%) = Ø
SET DSTAT% (QPT%) = 1
"DEVICE"; PNAME$(ABS(PTYPE%(QPT%)), 2); "STOPPED/DIABLED"
DRAWIT
COLOR SYMBOL QPT% BLACK
LOW LEVEL POINT? PTYPE%(QPT%) > Ø
YES
A
NO
SET I% = 1
E
I% = QPT%
YES
NO
B
C


B
IF ORIGINAL REQUEST WAS DISABLE THEN DISABLE ALL DISABLEABLE DEVICES BELOW.
IF ORIGINAL REQUEST WAS STOP, THE STOP ALL STOPPABLE DEVICES AND DISABLE ALL DISABLEABLE DEVICES BELOW.
I% POINT TIED TO OPT% (BASED ON PTIER%) ?
NO
YES
START/STOPPABLE? DEVICE. SSED%(I%) =1
YES
NO
WAS ORIGINAL REQUEST TO STOP? SSED%(OPT%) =1
NO
YES
SET DSTAT%(I%) =1
SET COBVL(I%) = Ø
"DEVICE"; PNAME$(ABS(PTYPE%(I%)),2) "DISABLED";
"DEVICE"; PNAME$(ABS(PTYPE%(I%)),2); " STOPPED";
ON SCREEN LPNT%(I%) = DIAMOND ?
NO
YES
DRAWIT
COLOR SYMBOL I% BLACK
C
INCREMENT I%
D


D
I% > NRPTS% ?
NO
E
YES
A
ALARM POINT? QPT% = APT%
NO
YES
ADISABLE
DISABLE ALARM
DRAWIT
DESELECT ORIGINALLY SELECTED POINT. QPT%
SET QPT% = 0
RETURN

NAME: XDD

PURPOSE:

The XDD subroutine is used to execute the DISPLAY DIAGRAM command function.

OPERATIONAL DESCRIPTION:

A filename is constructed which corresponds to a disk file containing the buffer memory image of the selected data environment diagram. This buffer file contains the static portion of the graphic display (e.g., the lines connecting the symbols). This file is loaded into a memory buffer and then displayed on the screen. DRAWIT is then used to place the correct symbols on the diagram using appropriate colors, rotation, etc. Colors and flashing mode are selected on the basis of the values stored for each point in the real-time data base.

CALLED BY:

CANCEL
CONFIRM
HELP

CALLS:

DRAWIT

PASSED ARGUMENTS:

APT%	- Current alarm point
CDBVL	- Real-time data bbse value for Ith point
FDEAM%	- Auto/manual indicator for DE
DIAMEN%	- Number of diagram or menu currently displayed
DSTAT%	- Disable status flag for Ith point

NAME: XDD (continued)

PASSED ARGUMENTS: (concluded)

LPNT%	- Diagram on which Ith point is located
NRTPTS%	- Number of real-time points
QFCN%	- Selected function
QMENU%	- Selected item
SSED%	- Start/Stop - Enable/Disable device discriminator

RETURNED ARGUMENTS:

DIAMEN%	- Number of diagram or menu currently displayed
FDEAM%	- Auto/Manual Indicator for DE
GDTYPE%	- Graphics display type
LDE%	- Last DE displayed
QFCN%	- Selected Function
QMENU%	- Selected menu item

FILE INPUT/OUTPUT:

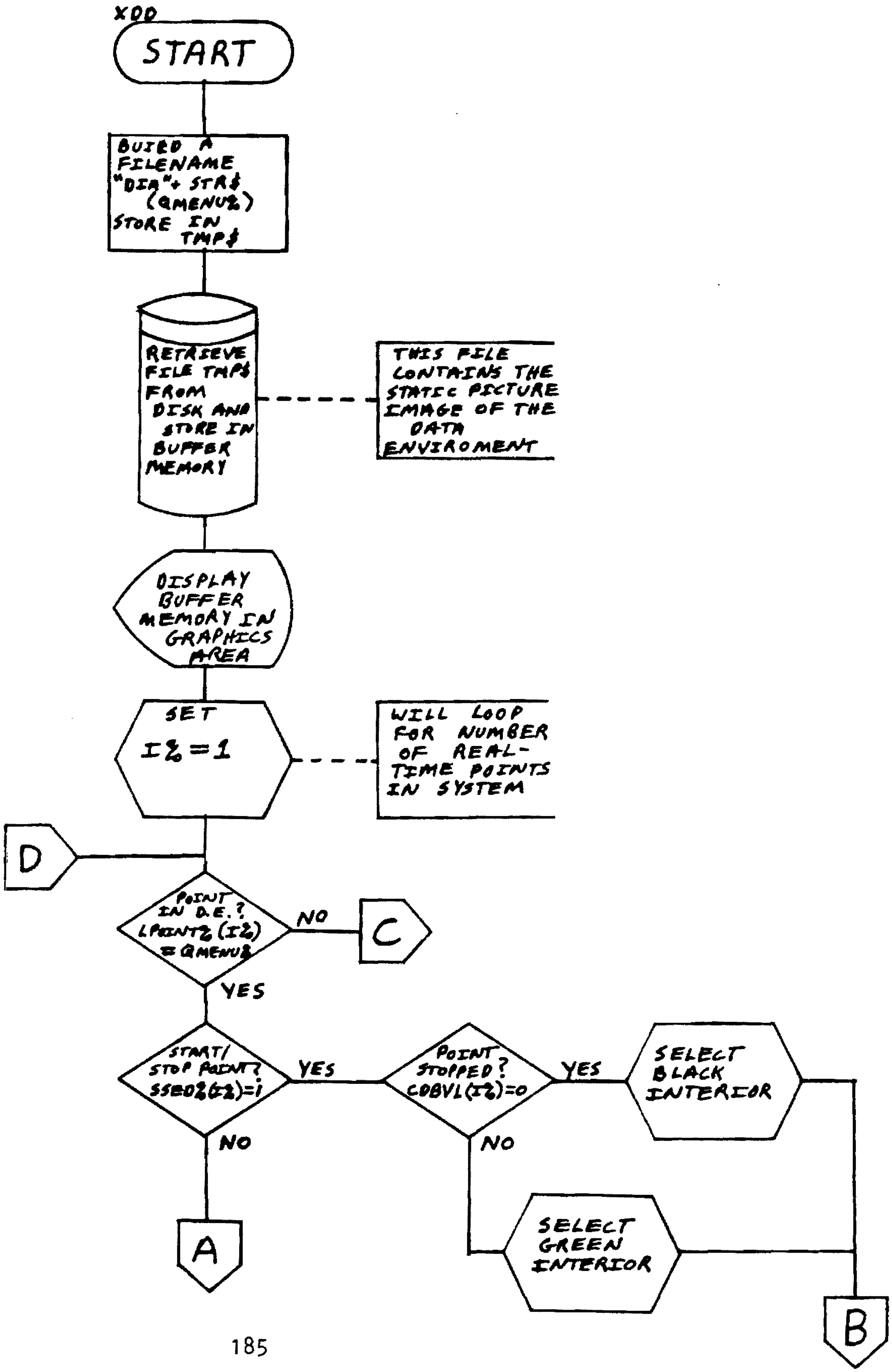
Files are retrieved from disk and loaded into memory for subsequent display on the screen. Filenames are of the form DIA*, where * is the number of the data environment to be displayed.

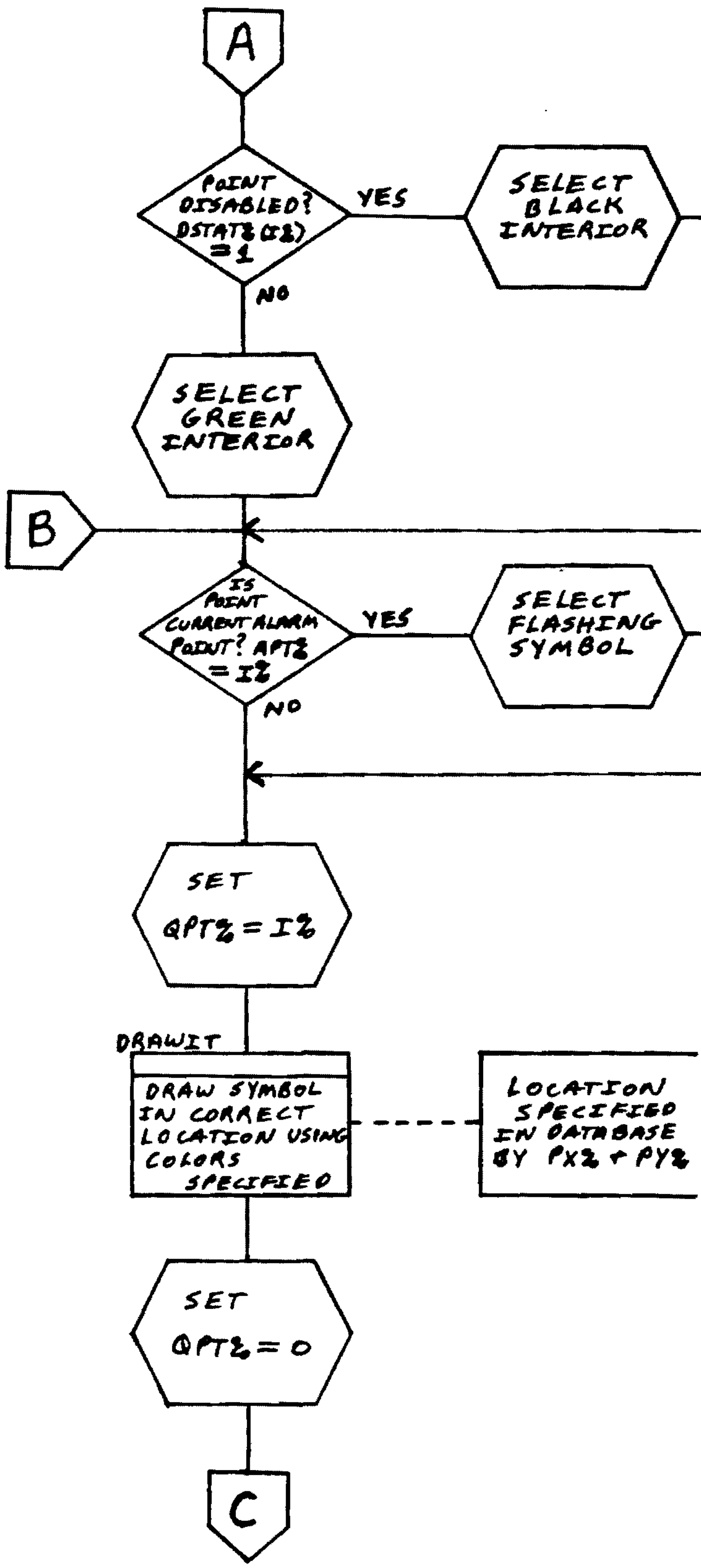
HARDWARE INTERACTION:

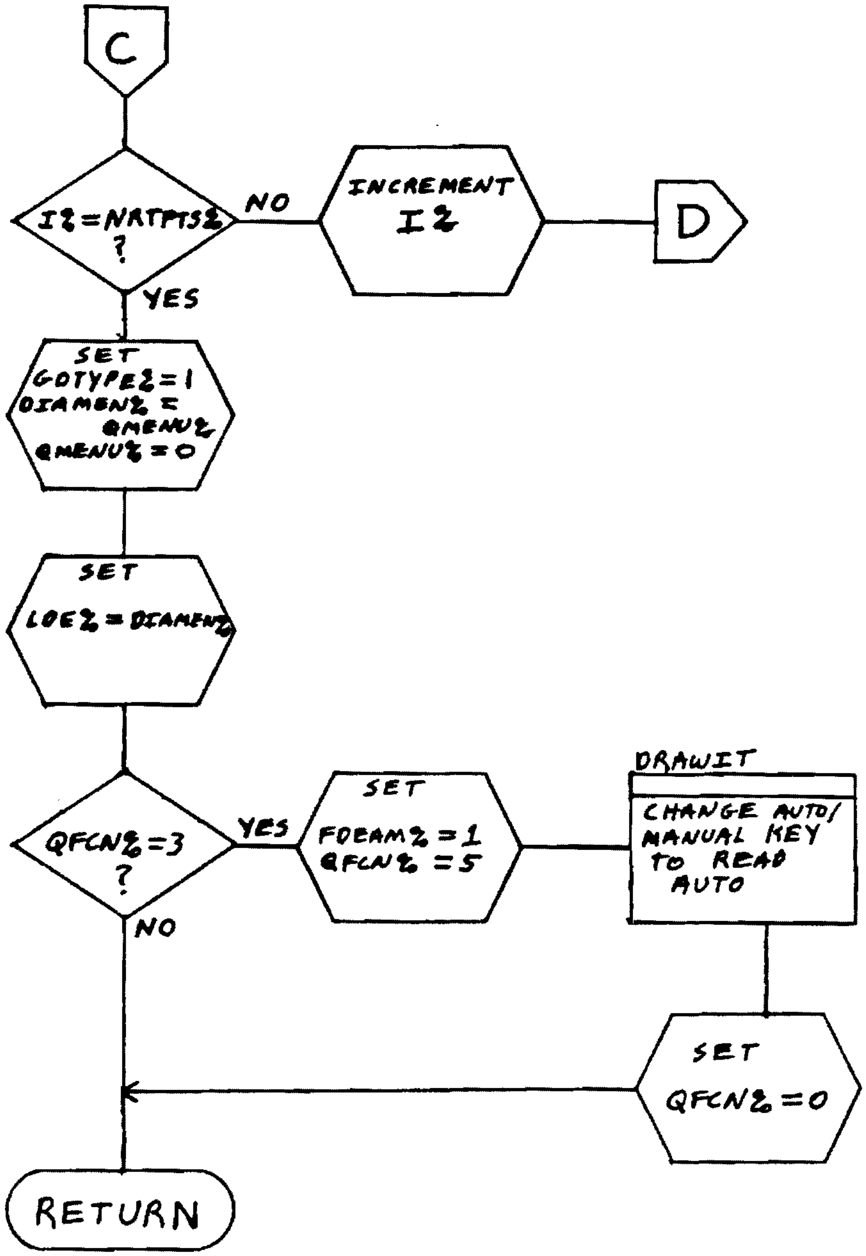
CRT DISPLAY - Direct manipulation of graphics area.

DESIGN NOTES:

(none)


XDD
START
BUILD A FILENAME "DIA"+ STR$ (QMENU%) STORE IN TMP$
RETRIEVE FILE TMP$ FROM DISK AND STORE IN BUFFER MEMORY
THIS FILE CONTAINS THE STATIC PICTURE IMAGE OF THE DATA ENVIROMENT
DISPLAY BUFFER MEMORY IN GRAPHICS AREA
SET I% = 1
WILL LOOP FOR NUMBER OF REAL-TIME POINTS IN SYSTEM
D
POINT IN D.E.? LPOINT%(I%) = QMENU%
NO
C
YES
START/ STOP POINT? SSED%(I%)=1
YES
POINT STOPPED? CDBVL(I%)=0
YES
SELECT BLACK INTERIOR
NO
NO
SELECT GREEN INTERIOR
A
B


A
POINT DISABLED? DSTAT%(I%) = 1
YES
SELECT BLACK INTERIOR
NO
SELECT GREEN INTERIOR
B
IS POINT CURRENT ALARM POINT? APT% = I%
YES
SELECT FLASHING SYMBOL
NO
SET QPT% = I%
DRAWIT
DRAW SYMBOL IN CORRECT LOCATION USING COLORS SPECIFIED
LOCATION SPECIFIED IN DATABASE BY PX% + PY%
SET QPT% = 0
C


C
I% = NRTPTS% ?
NO
INCREMENT I%
D
YES
SET
GOTYPE% = 1
DIAMEN% = QMENU%
QMENU% = 0
SET
LOE% = DIAMEN%
QFCN% = 3 ?
YES
SET
FOEAM% = 1
QFCN% = 5
DRAWIT
CHANGE AUTO/ MANUAL KEY TO READ AUTO
NO
SET
QFCN% = 0
RETURN

NAME: XSPL

PURPOSE:

The XSPL subroutine is used to execute the SET POINT/LIMITS function.

OPERATIONAL DESCRIPTION:

The XPSL subroutine replaces the current data base values for the low limit, high limit, and set point of the selected point with those stored in the temporary storage variables LTMP, HTMP, and STMP, respectively. If the point is the current alarm point, then the ADISABLE module is called to disable the alarm.

CALLED BY:

CONFIRM

CALLS:

DISMESS
ADISABLE

PASSED ARGUMENTS:

APT%	- Current alarm point
HTMP	- New analog high limit
LTMP	- New analog low limit
QPT%	- Selected point
PNAME$	- Textual name of point
PTYPE%	- Point type for Ith point
STMP	- New analog set point

NAME: XSPL

RETURNED ARGUMENTS:

HILIM	- Current analog high limit
LOWLIM	- Current analog low limit
QPT%	- Selected point
SPT	- Current analog set point

FILE INPUT/OUTPUT:

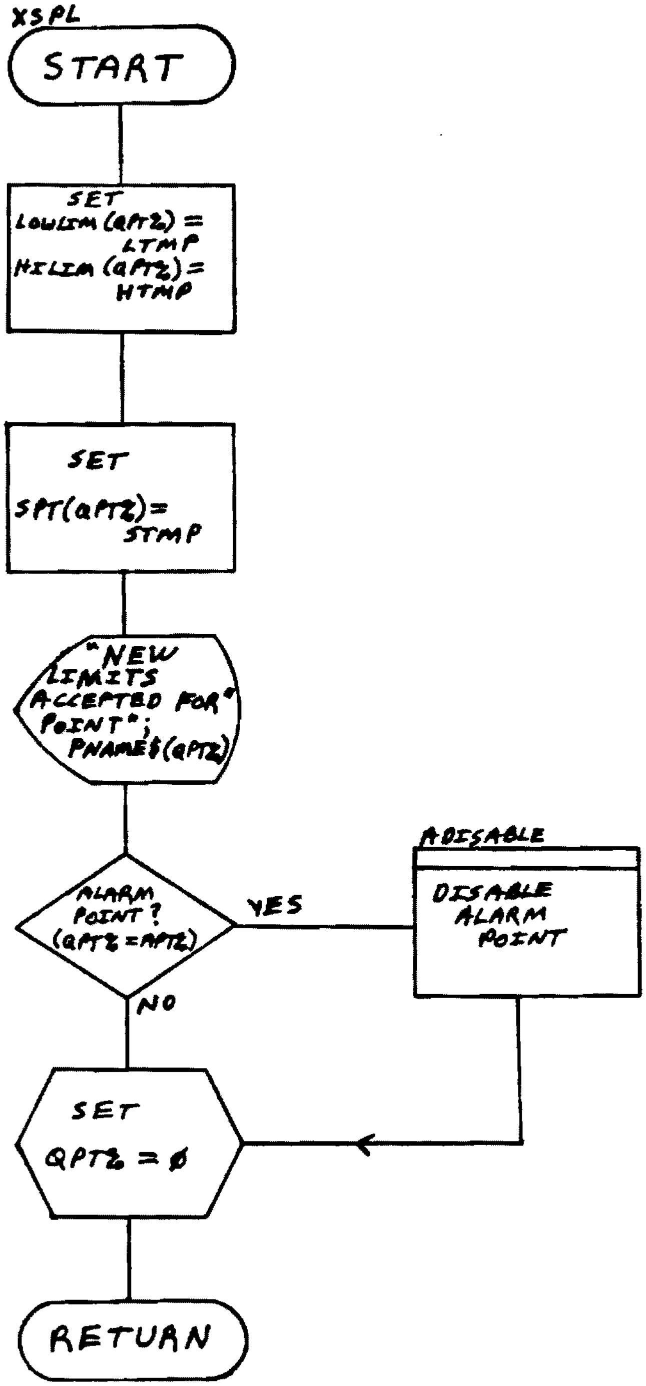
(none)

HARDWARE INTERACTION:

CRT DISPLAY - prints messages to text area.

DESIGN NOTES:

The temporary storage variables hold the new values retrieved by the MLIN from the keyboard. These values will be stored as new operational values when CONFIRM ACTION is touched.


XSPL
START
SET
LOWLIM (QPT%) = LTMP
HILIM (QPT%) = HTMP
SET
SPT(QPT%) = STMP
"NEW LIMITS ACCEPTED FOR POINT"; PNAME$(QPT%)
ALARM POINT? (QPT% = APT%)
YES
NO
ADISABLE
DISABLE ALARM POINT
SET
QPT% = Ø
RETURN

NAME: XAM

PURPOSE:

The XAM subroutine executes the AUTO/MANUAL function.

OPERATIONAL DESCRIPTION:

XAM updates the screen and the database to reflect the newly selected mode of operation.

CALLED BY:

CONFIRM

CALLS:

DRAWIT

PASSED ARGUMENTS:

DIAMEN%	- Number of diagram or menu currently displayed
QAM%	- Selected mode
TT$	- Textual name of DE

NAME: XAM (continued)

RETURNED ARGUMENTS:

FDEM%	-	Auto/manual mode indicator for DE
QAM%	-	Selected mode

FILE INPUT/OUTPUT:

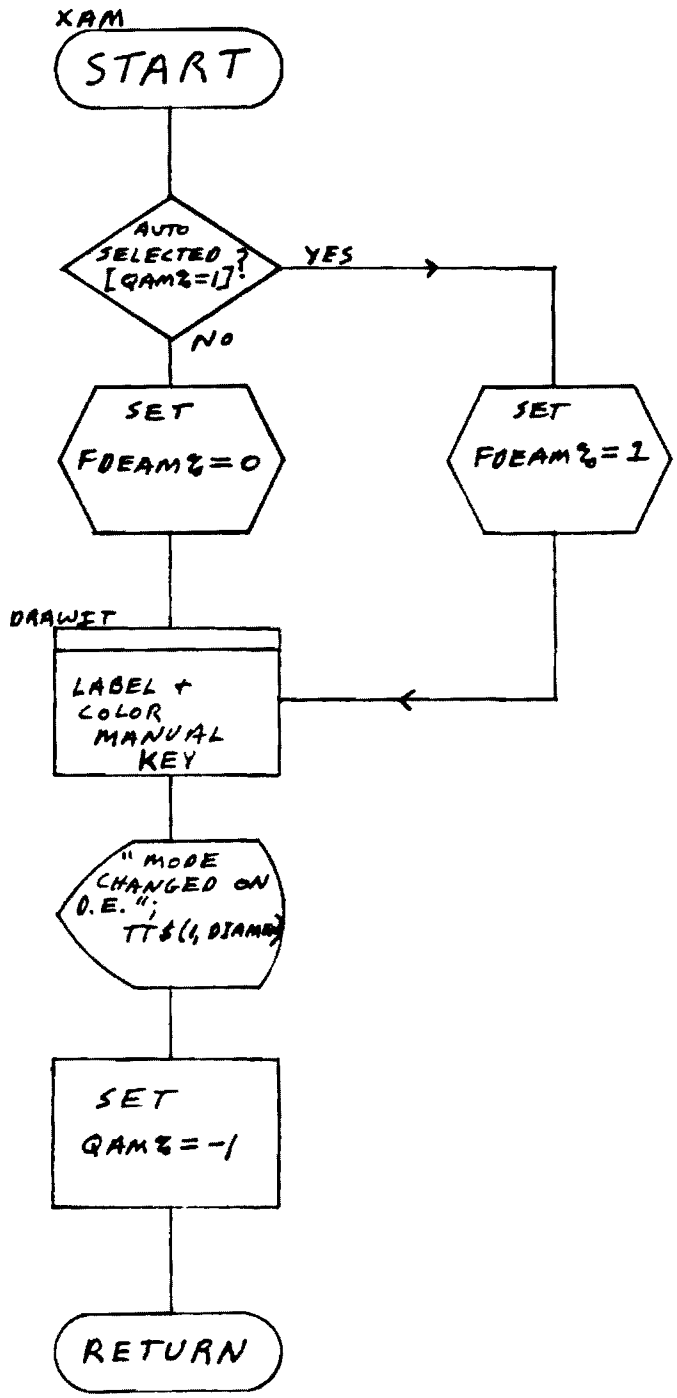
(none)

HARDWARE INTERACTION:

CRT DISPLAY - Print messages

DESIGN NOTES:

(none)


XAM
START
AUTO SELECTED ? [QAM%=1]?
YES
NO
SET FOEAM%=0
SET FOEAM%=1
DRAWIT
LABEL + COLOR MANUAL KEY
"MODE CHANGED ON O.E."; TT$(1,DIAM
SET QAM%=-1
RETURN

NAME: XPR

PURPOSE:

The XPR subroutine is used to execute the PRINT REPORT function.

OPERATIONAL DESCRIPTION:

A filename is constructed which corresponds to the disk file containing the desired report. This file is then retrieved and sent to the printer or CRT depending on the value of the report output device selection variable (RODS%).

CALLED BY:

CONFIRM

CALLS:

DISMESS

PASSED ARGUMENTS:

QMENU%	- Selected menu item
RODS%	- Report output device selection
ZIN$	- Raw keyboard input

RETURNED ARGUMENTS:

GDTYPE%	- Graphics display type
QMENU%	- Selected menu item
RODS%	- Report output device selection

NAME: XPR (continued)

FILE INPUT/OUTPUT:

Buffer files are retrieved from disk and sent to the printer or CRT screen. Filenames are of the form RPT*, where * is the number of the desired report.

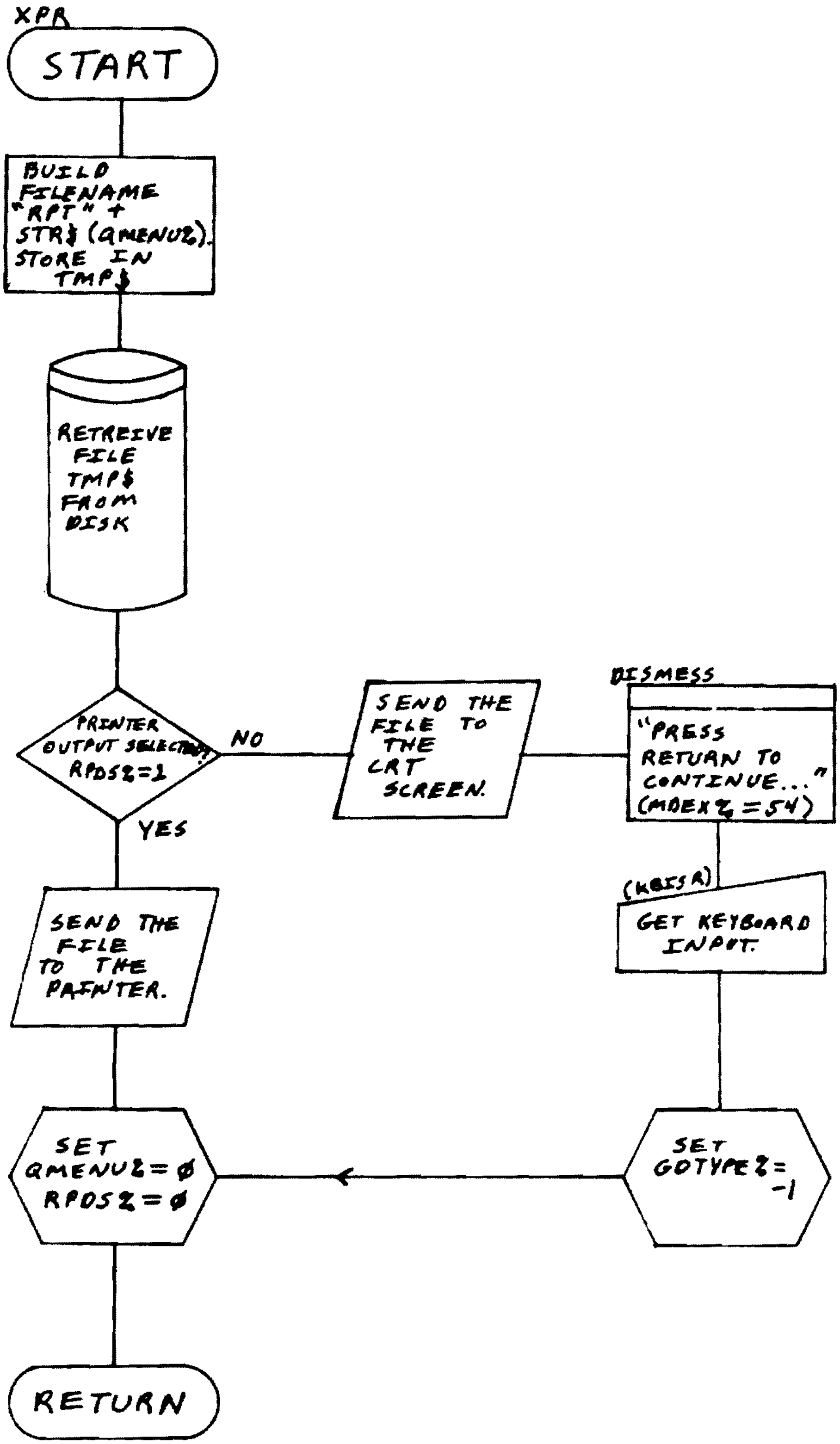
HARDWARE INTERACTION:

CRT - Display in graphics area.

PRINTER - Print reports

DESIGN NOTES:

(none)


XPR
START
BUILD FILENAME "RPT" + STR$ (QMENU%). STORE IN TMP$
RETRIEVE FILE TMP$ FROM DISK
PRINTER OUTPUT SELECTED? RPOS%=1
NO
YES
SEND THE FILE TO THE CRT SCREEN.
DISMESS
"PRESS RETURN TO CONTINUE..." (MDEX% = 54)
(KBIS R)
GET KEYBOARD INPUT.
SEND THE FILE TO THE PRINTER.
SET QMENU% = Ø RPOS% = Ø
SET GOTYPE% = -1
RETURN

NAME: XMS

PURPOSE:

The XMS subroutine is used to execute the MODIFY SCHED function.

OPERATIONAL DESCRIPTION:

The XMS subroutine replaces the current real-time data base schedule for the selected DE with the schedule as modified by the operator.

CALLED BY:

CONFIRM

CALLS:

(none)

PASSED ARGUMENTS:

NSE%	- Number of schedule entries
QMENU%	- Selected menu item
QSCH%	- Number of the data environment schedule to be modified
SCHTMP$	- Temporary storage for new (modified) schedule
TT$	- Text strings for DE's

RETURNED ARGUMENTS:

S$	- Current real-time schedule for DE
QMENU%	- Selected menu item
QSCH%	- Number of DE schedule to be modified
ZC%	- Internal control variable

NAME: XMS (continued)

FILE INPUT/OUTPUT:

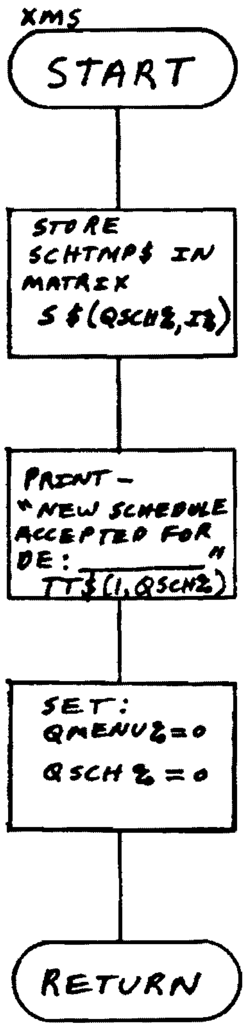
(none)

HARDWARE INTERACTION:

CRT Screen - Print message in text area

DESIGN NOTES:

(none)


XMS
START
STORE SCHTMP$ IN MATRIX S$(QSCH%,I%)
PRINT - "NEW SCHEDULE ACCEPTED FOR DE: ____" TT$(1,QSCH%)
SET: QMENU%=0 QSCH%=0
RETURN

NAME: XCHOP

PURPOSE:

The XCHOP subroutine is used to execute the CHANGE OPER function.

OPERATIONAL DESCRIPTION:

Based upon the value of QMENU%, XCHOP calls either the HELP subroutine, sets the shutdown control variable, or performs the change operator operation. If the change operator action has been selected, then XCHOP requests the operator to type in name and password. If the name and password are valid, XCHOP updates the Date/Time Operator display to reflect the new operator's name. Otherwise appropriate error messages are given.

CALLED BY:

CONFIRM

CALLS:

DISMESS
HELP
(KB ISR)

PASSED ARGUMENTS:

NOPER%	- Number of system operators
OPER$	- String array containing operator's name
PWRD$	- String array containing operator's password
QMENU%	- Selected menu item
ZIN$	- Raw keyboard input

NAME: XCHOP (continued)

RETURNED ARGUMENTS:

OPER% - Number of the current system operator

SHUTDOWN% - Flag to stop system

FILE INPUT/OUTPUT:

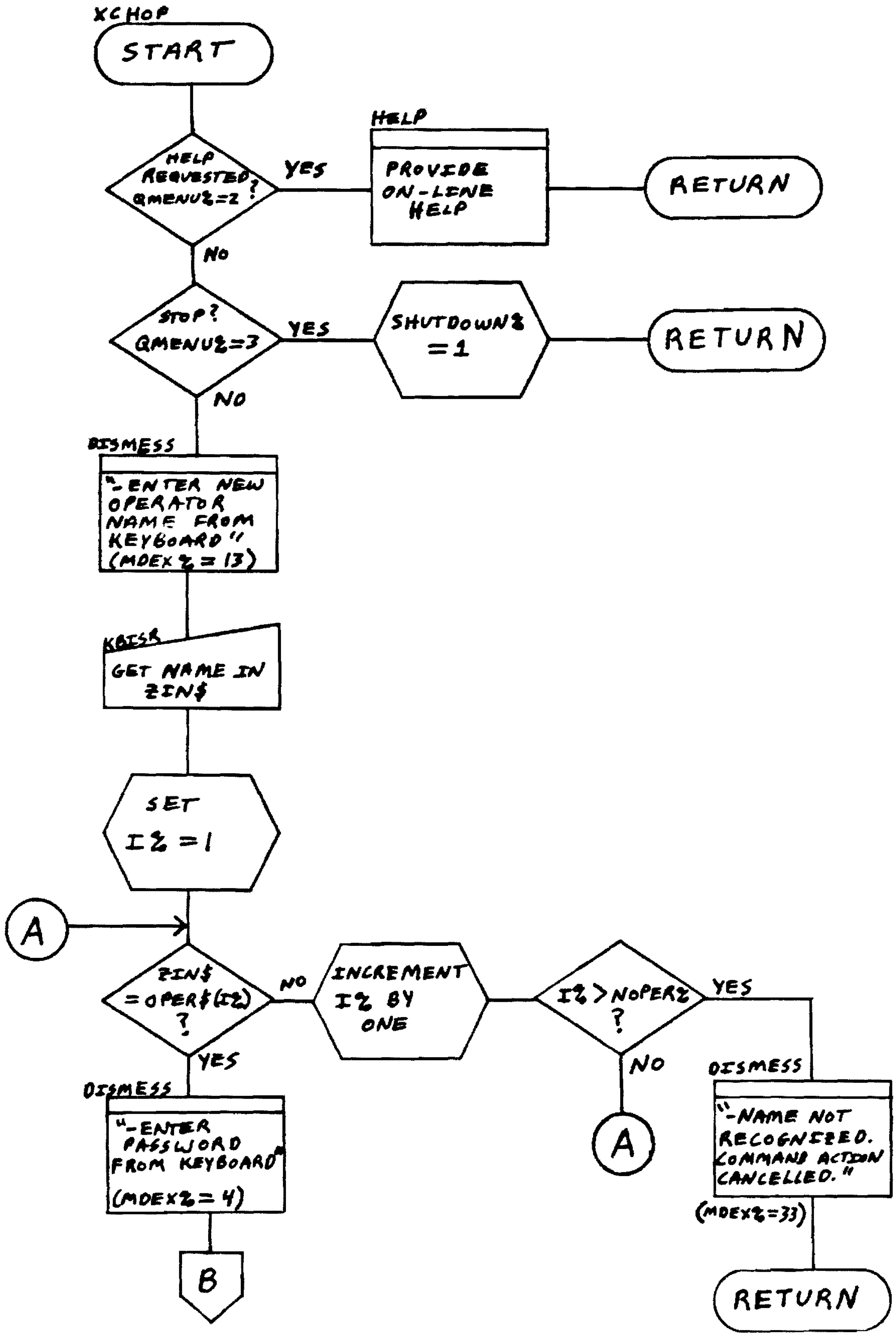
(none)

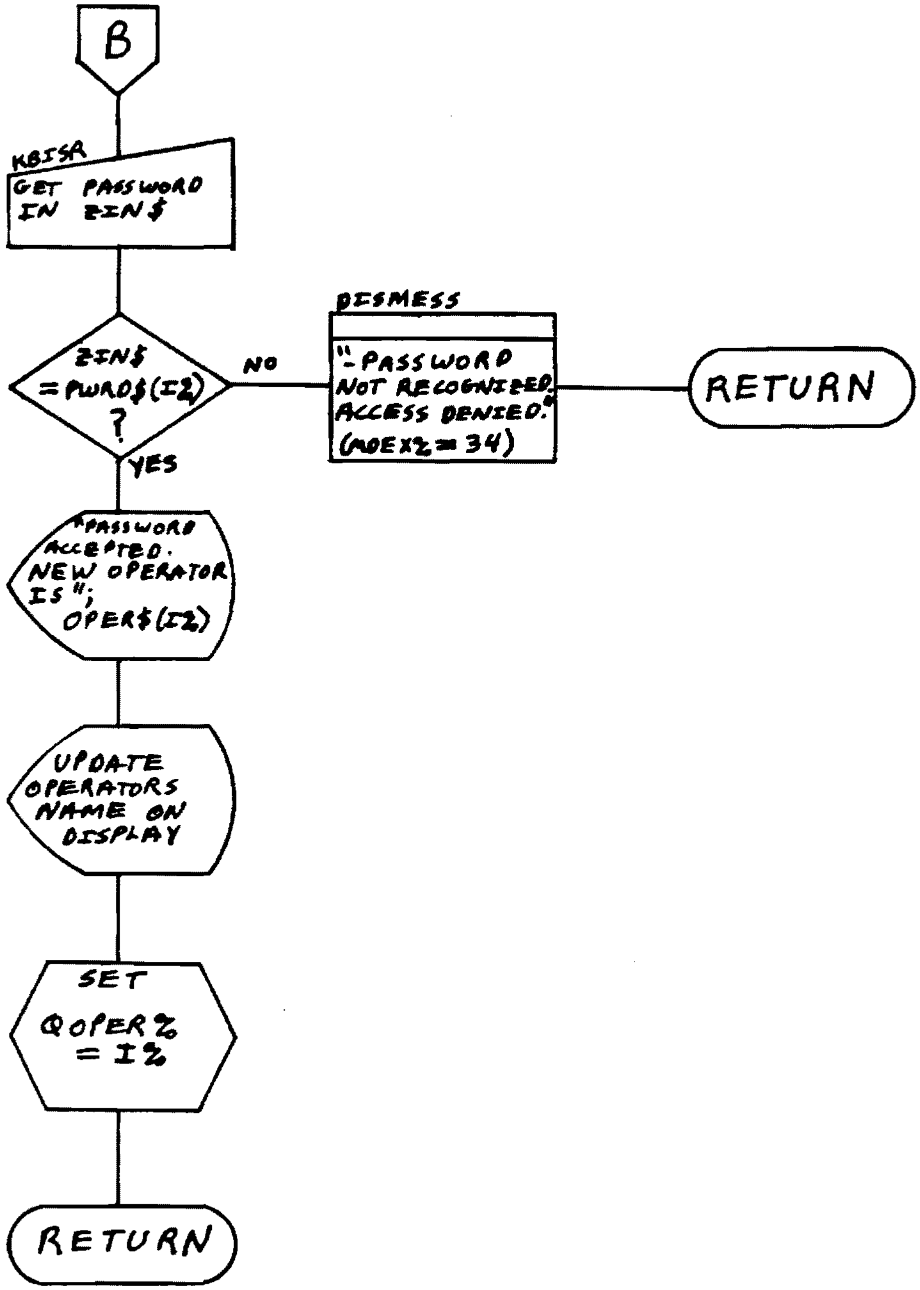
HARDWARE INTERACTION:

CRT Screen - Displays messages in text area. Also, Direct manipulation of Date/Time/Operator display area.

DESIGN NOTES:

(none)


XCHOP
START
HELP REQUESTED? QMENU%=2 ?
YES
HELP
PROVIDE ON-LINE HELP
RETURN
NO
STOP? QMENU%=3
YES
SHUTDOWN% =1
RETURN
NO
DISMESS
"-ENTER NEW OPERATOR NAME FROM KEYBOARD" (MDEX% = 13)
KBISR
GET NAME IN ZIN$
SET I% = 1
A
ZIN$ = OPER$(I%) ?
NO
INCREMENT I% BY ONE
I% > NOPER% ?
YES
NO
A
DISMESS
"-NAME NOT RECOGNIZED. COMMAND ACTION CANCELLED."
(MDEX%=33)
RETURN
YES
DISMESS
"-ENTER PASSWORD FROM KEYBOARD"
(MDEX% = 4)
B


B
KBISR
GET PASSWORD IN ZIN$
ZIN$ = PWRD$(I%) ?
NO
DISMESS
"-PASSWORD NOT RECOGNIZED. ACCESS DENIED." (MOEX% = 34)
RETURN
YES
"PASSWORD ACCEPTED. NEW OPERATOR IS"; OPER$(I%)
UPDATE OPERATORS NAME ON DISPLAY
SET QOPER% = I%
RETURN

NAME: DISMESS

PURPOSE:

The DISMESS subroutine is used to display color coded messages from the master message file in the message/text area of the CRT display unit.

OPERATIONAL DESCRIPTION:

The DISMESS subroutine first gets the color coded message text from the master message file on disk using MDEX% as the pointer to the record containing the desired message. The message is printed in the text area and program control returns to the calling routine.

CALLED BY:

ANNUNC
AUTO
BUZOFF
CANCEL
CONFIRM
DIAGRAM
DSCHED
FKEYIN
GONOGO
HELP
MLIN
MOIN
MRIN
MSIN
MMI EXEC
MSCHED
OPER
REPORT
REPOUT
SCHED

NAME: DISMESS (continued)

CALLED BY: (continued)

SETPT

SYMBIN

XPR

CALLS:

(none)

PASSED ARGUMENTS:

MDEX% - Message/record number

RETURNED ARGUMENTS:

(none)

FILE INPUT/OUTPUT:

Read message text from random access disk file, MESSAGE.DAT.

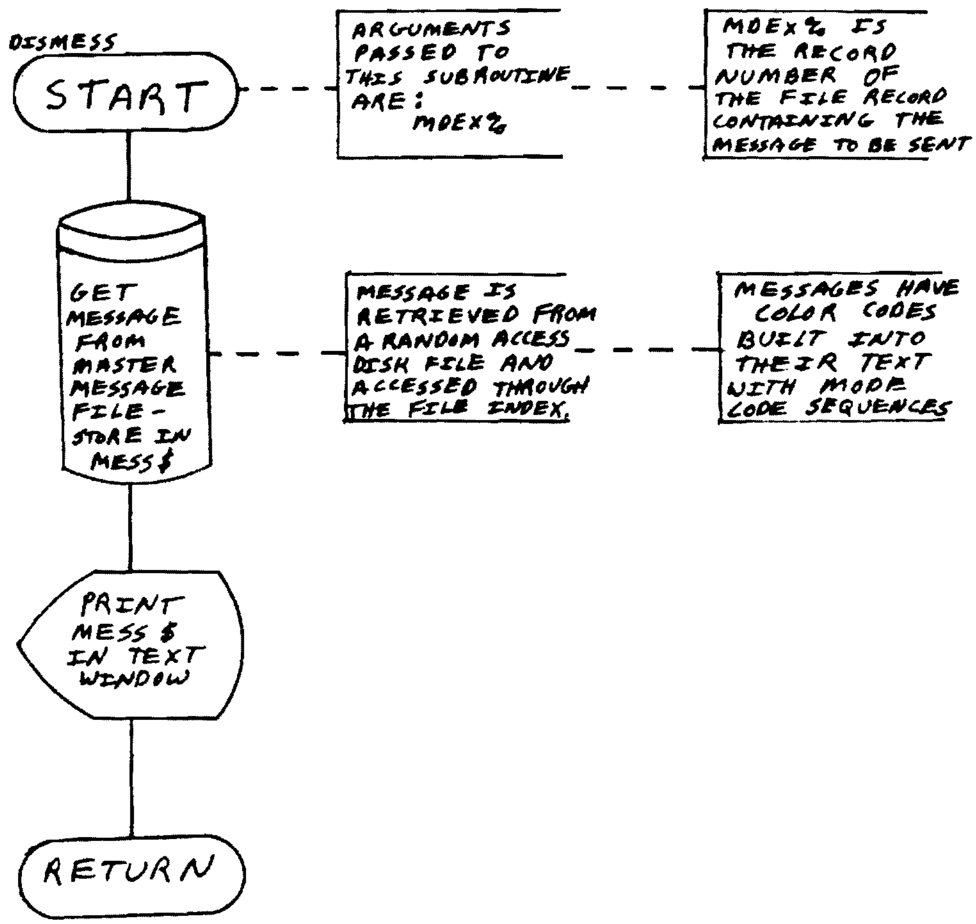
HARDWARE INTERACTION:

CRT SCREEN - Displays messages in text area.

DESIGN NOTES:

(none)

DISMESS : DISPLAY MESSAGES IN THE TEXT AREA FROM THE MASTER MESSAGE FILE



NAME: DMENU

PURPOSE:

The DMENU subroutine is used to place the appropriate menu on the screen.

OPERATIONAL DESCRIPTION:

The DMENU subroutine places the menu title on the screen. If the menu is for the SET POINT/LIMITS function then temporary storage for the high limit, low limit, and set point is set up. DRAWIT is then called to display the menu. Program control then returns to the calling routine.

CALLED BY:

CONFIRM
DIAGRAM
OPER
REPORT
SCHED
SETPT
SYMBIN

CALLS:

DRAWIT

NAME: DMENU (concluded)

PASSED ARGUMENTS:

APT%	- Current alarm point
DIAMEN%	- Number of diagram or menu currently displayed
HILIM	- High limit for analog point
LOWLIM	- Low limit for analog point
LPNT%	- Process diagram on which Ith point is located
QFCN%	- Selected function
NBOXES%	- Number of menu boxes for menus
QPT%	- Seleted point
SPT	- Set point for analog point
TT$	- Textual names for DE's

RETURNED ARGUMENTS:

GDTYPE%	- Graphics Display Type
HTMP	- High limit temporary storage
LTMP	- Low limit temporary storage
QMENU%	- Selected Menu Item
STMP	- Set Point temporary storage

FILE INPUT/OUTPUT:

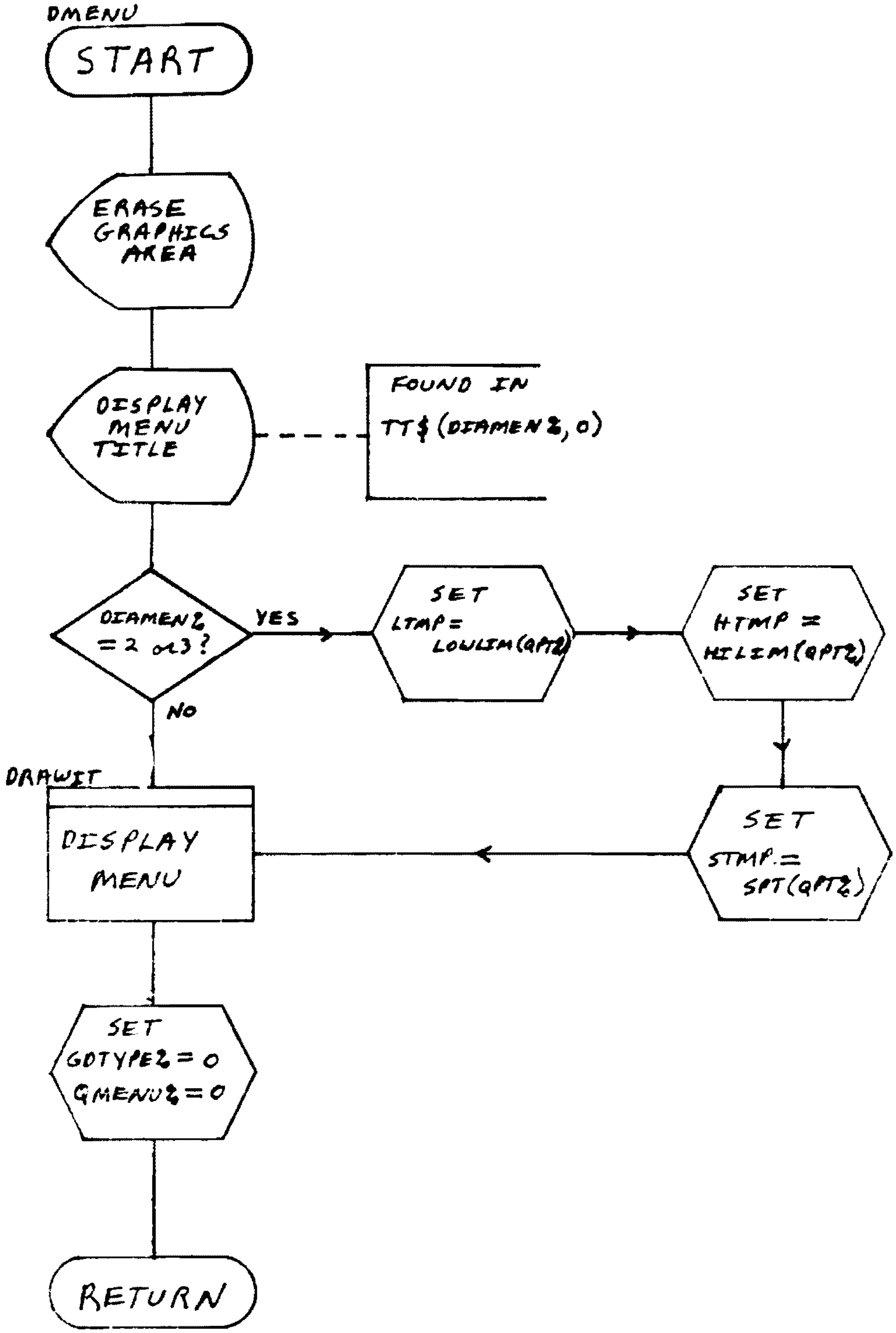
(none)

HARDWARE INTERACTION:

(none)

DESIGN NOTES:

(none)


DMENU
START
ERASE GRAPHICS AREA
DISPLAY MENU TITLE
FOUND IN
TT$(DIAMEN%,0)
DIAMEN% = 2 or 3?
YES
NO
SET LTMP = LOWLIM(QPT%)
SET HTMP = HILIM(QPT%)
SET STMP = SPT(QPT%)
DRAWIT
DISPLAY MENU
SET GDTYPE% = 0 GMENU% = 0
RETURN

NAME: DSCHED

PURPOSE:

Display selected schedule

OPERATIONAL DESCRIPTION:

The DSCHED module sets up temeporary storage for the selected schedule to be modified. Next, it displays the static portion of the schedule layout, which it retrieves from disk. Then DSCHED displays the schedule name and entries on the schedule layout.

CALLED BY:

CONFIRM

CALLS:

DISMESS

DRAWIT

PASSED ARGUMENTS:

NSE%	- Number of schedule entries
QFCN%	- Selected function
QSCH%	- Selected schedule
S$	- Schedules for all data environments
TT$	- Text descriptors for DE's
VPS%	- Schedule entry coordinates

RETURNED ARGUMENTS:

QFCN%	- Selected function
SCHTMP$	- Currently selected schedule to be modified

NAME: DSCHED (continued)

FILE INPUT/OUTPUT:

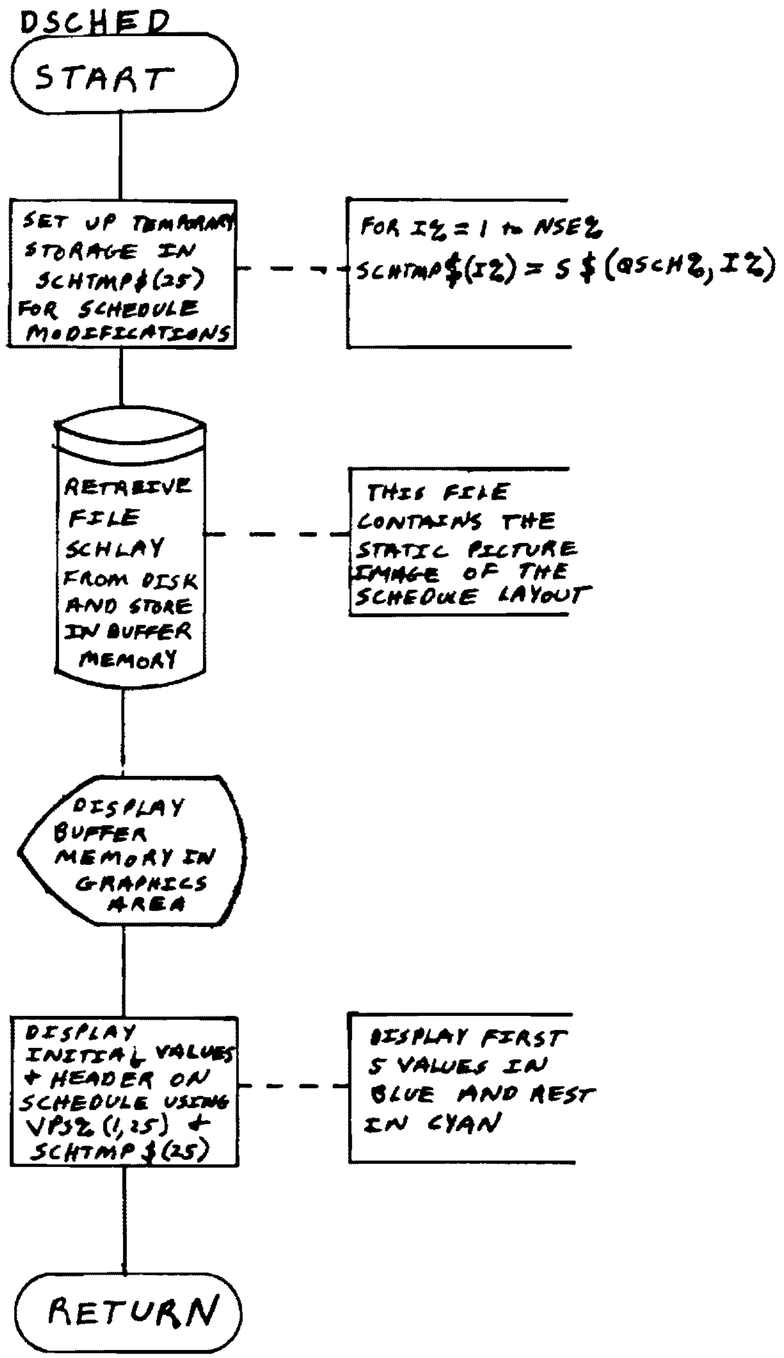
Retrieves static image of schedule layout from disk file SCHLAY.BUF.

HARDWARE INTERACTION:

CRT Screen - Direct manipulation of graphics area.

DESIGN NOTES:

(none)


DSCHED
START
SET UP TEMPORARY STORAGE IN SCHTMP$(25) FOR SCHEDULE MODIFICATIONS
FOR I% = 1 to NSE%
SCHTMP$(I%) = S$(@SCH%, I%)
RETRIEVE FILE SCHLAY FROM DISK AND STORE IN BUFFER MEMORY
THIS FILE CONTAINS THE STATIC PICTURE IMAGE OF THE SCHEDULE LAYOUT
DISPLAY BUFFER MEMORY IN GRAPHICS AREA
DISPLAY INITIAL VALUES + HEADER ON SCHEDULE USING VPS%(1,25) + SCHTMP$(25)
DISPLAY FIRST 5 VALUES IN BLUE AND REST IN CYAN
RETURN

NAME: DRAWIT

PURPOSE:

The DRAWIT utility subroutine is used to draw and redraw HVAC symbols, function keys, menu items, and the alarm indicator. Parameters are used to select interior color, border color, flashing, and rotation.

OPERATIONAL DESCRIPTION:

The DRAWIT subroutine selects the options specified by the parameters and uses the graphics capability of the computer to draw the object selected on the screen.

CALLED BY:

ANNUNC
AUTO
CANCEL
CONFIRM
DIAGRAM
DSCHED
FKEYIN
GONOGO
MDIN
MLIN
MRIN
MSIN
OPER
REPORT
SCHED
SETPT
SYMBIN
XAM
XDD

NAME: DRAWIT (continued)

CALLED BY: (concluded)

XSD

XSE

XSPL

CALLS:

(none)

PASSED ARGUMENTS:

D1%	- =-1 menu box	
	- = 0 function key box	
	- = 1 symbol	
	- = 2 alarm indicator on	
	- = 3 alarm indicator off	
D2%	- = 1 change interior	
	- = 2 change border	
	- = 3 change both	
D3%	- = 1 yellow border (else blue or cyan)	
	- = 2 green interior (else black)	
	- = 8 blink (else non-blink)	
DIAMEN%	- Number of diagram or menu currently displayed	
FDEAM%	- Auto/manual indicator for DE	
GDTYPE%	- Graphics Display Type	
LALARM	- Alarm (monitor) point location in SID$	
LCNT%	- Control point location in SID$	
LIND%	- Indicator point location in SID$	
KCOLR%	- Function key colors	
KTEXT$	- Function key text strings	
NALARM%	- Number of alarm point types	
NCNT%	- Number of control point types	
NIND%	- Number of indicator point types	

NAME: DRAWIT (continued)

PASSED ARGUMENTS: (concluded)

PTYPE%	- Point Type
PX%	- X coordinate for points
PY%	- Y coordinate for points
QFCN%	- Selected function
QMENU%	- Selected menu item
QPT%	- Currently selected point
SID$	- Symbol identifier string
SL%	- Symbol identifier string length
TT$	- Textual names for DE's

RETURNED ARGUMENTS:

(none)

FILE INPUT/OUTPUT:

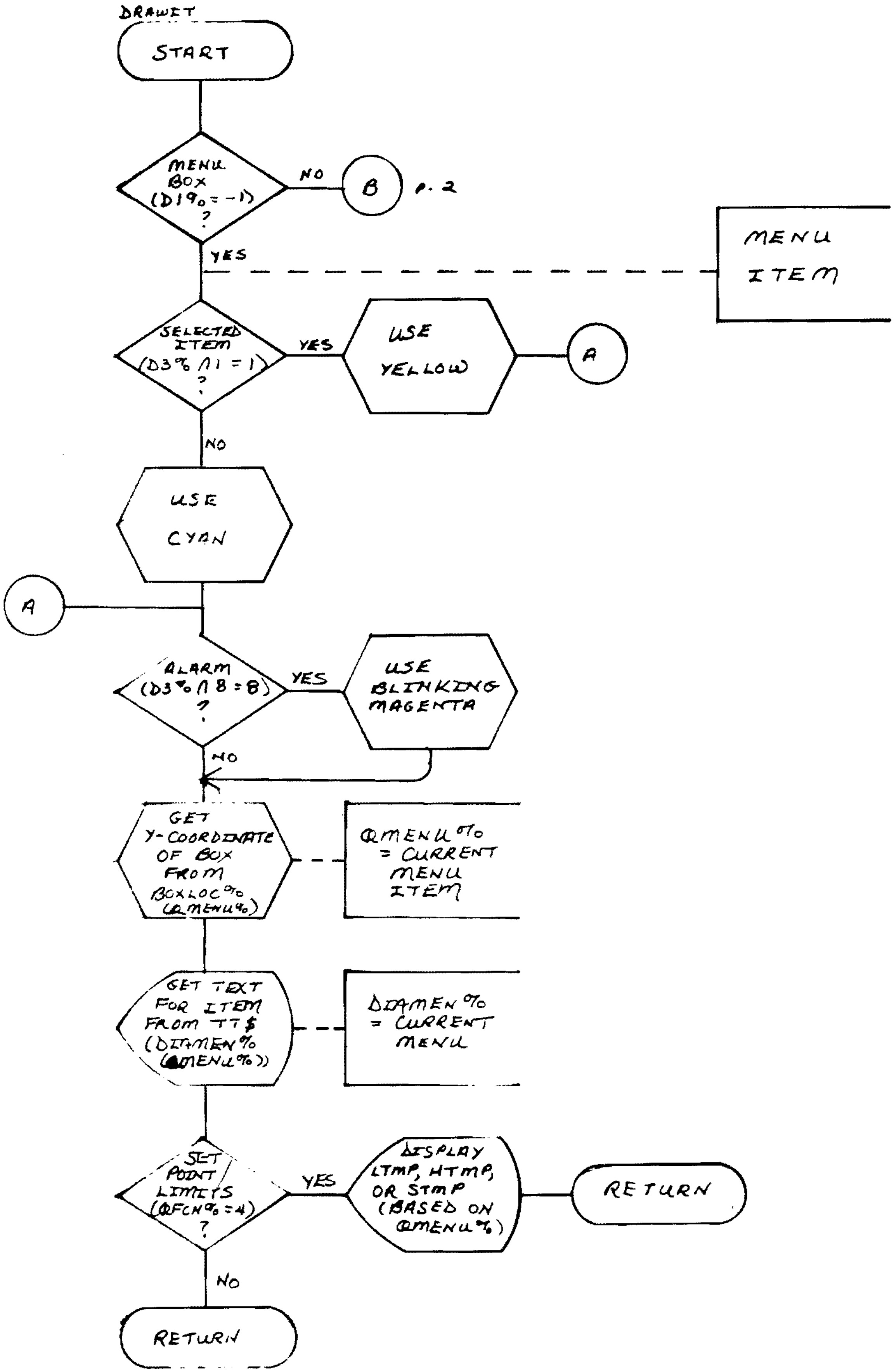
(none)

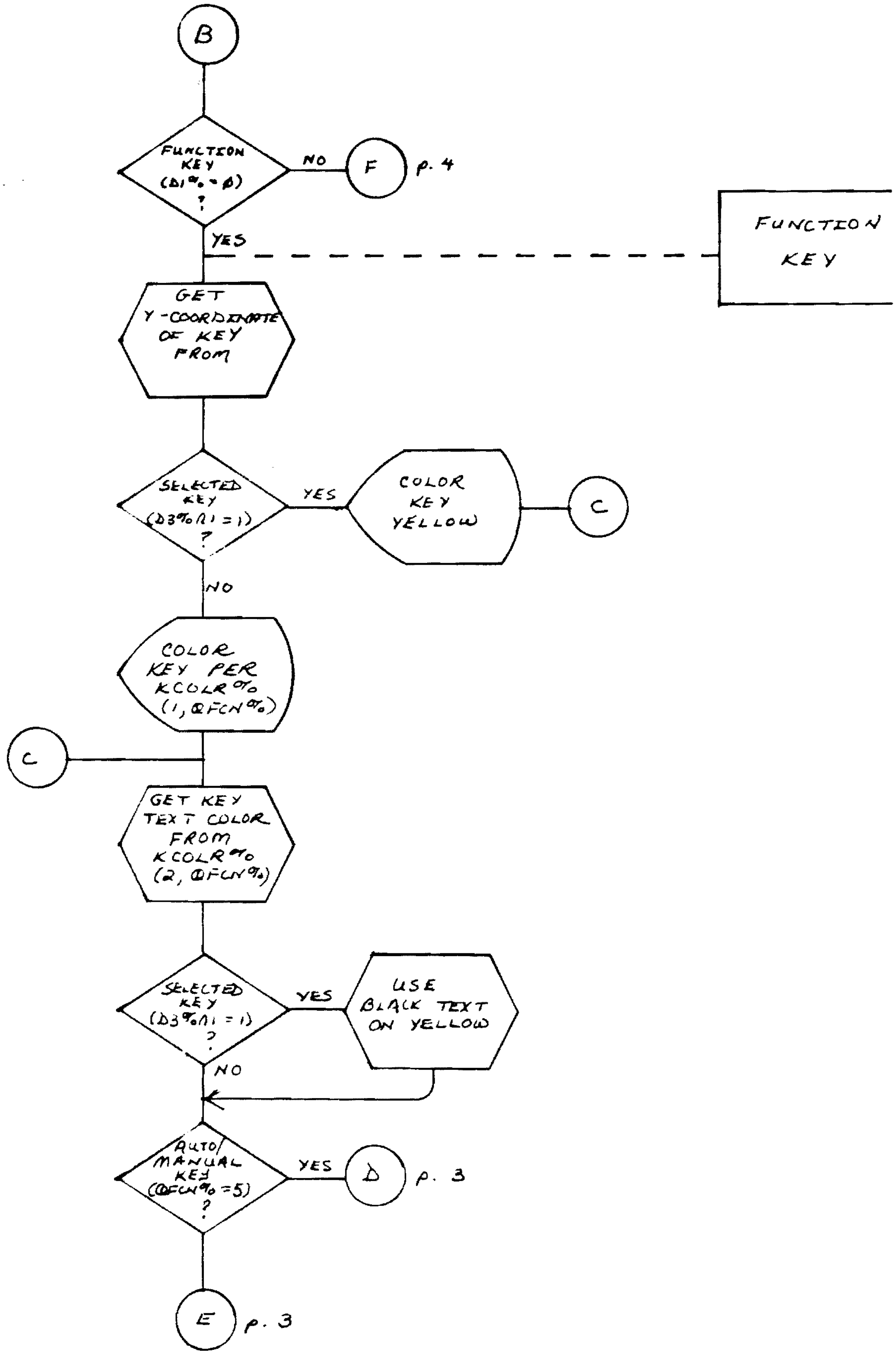
HARDWARE INTERACTION:

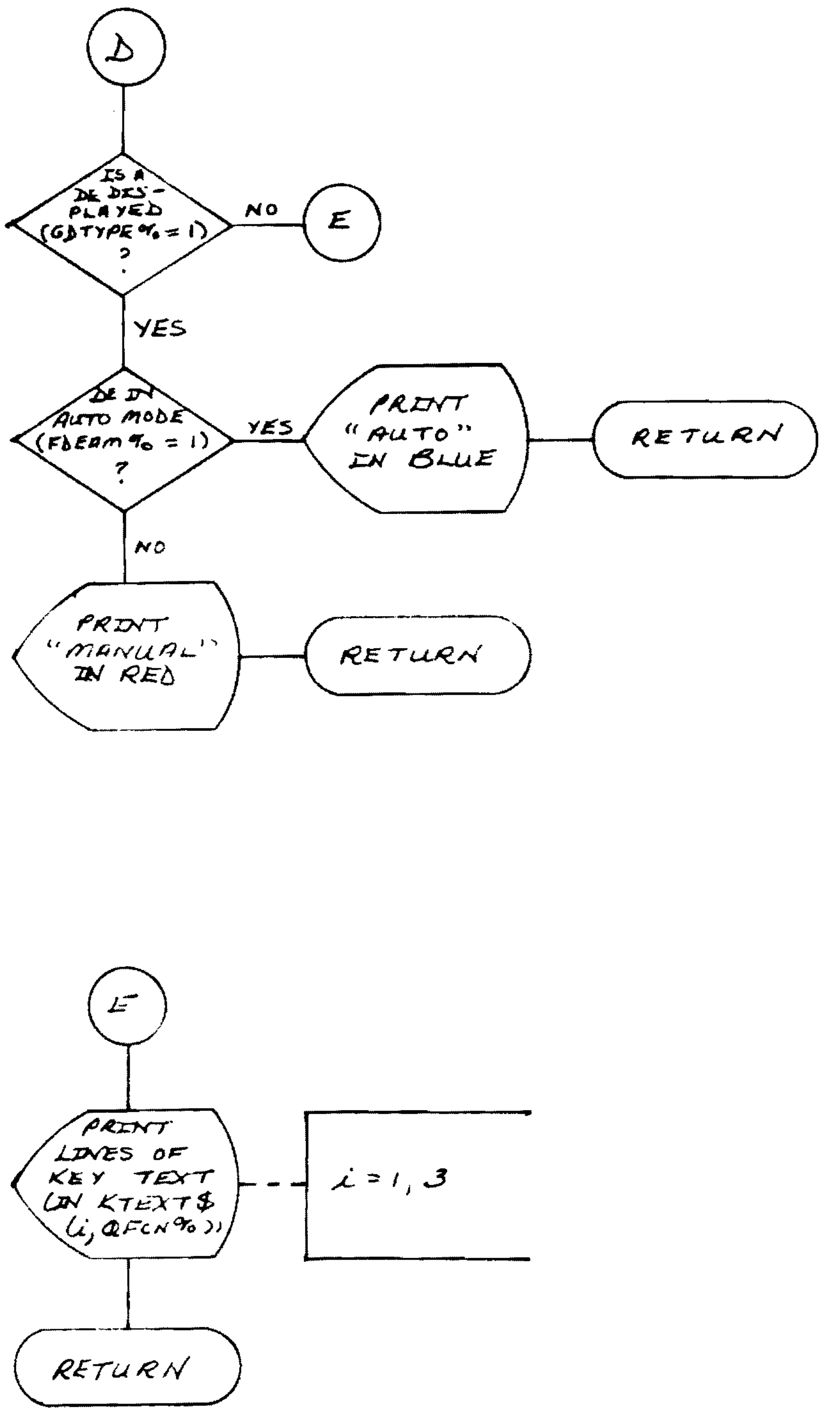
CRT DISPLAY - Direct manipulation of CRT screen display.

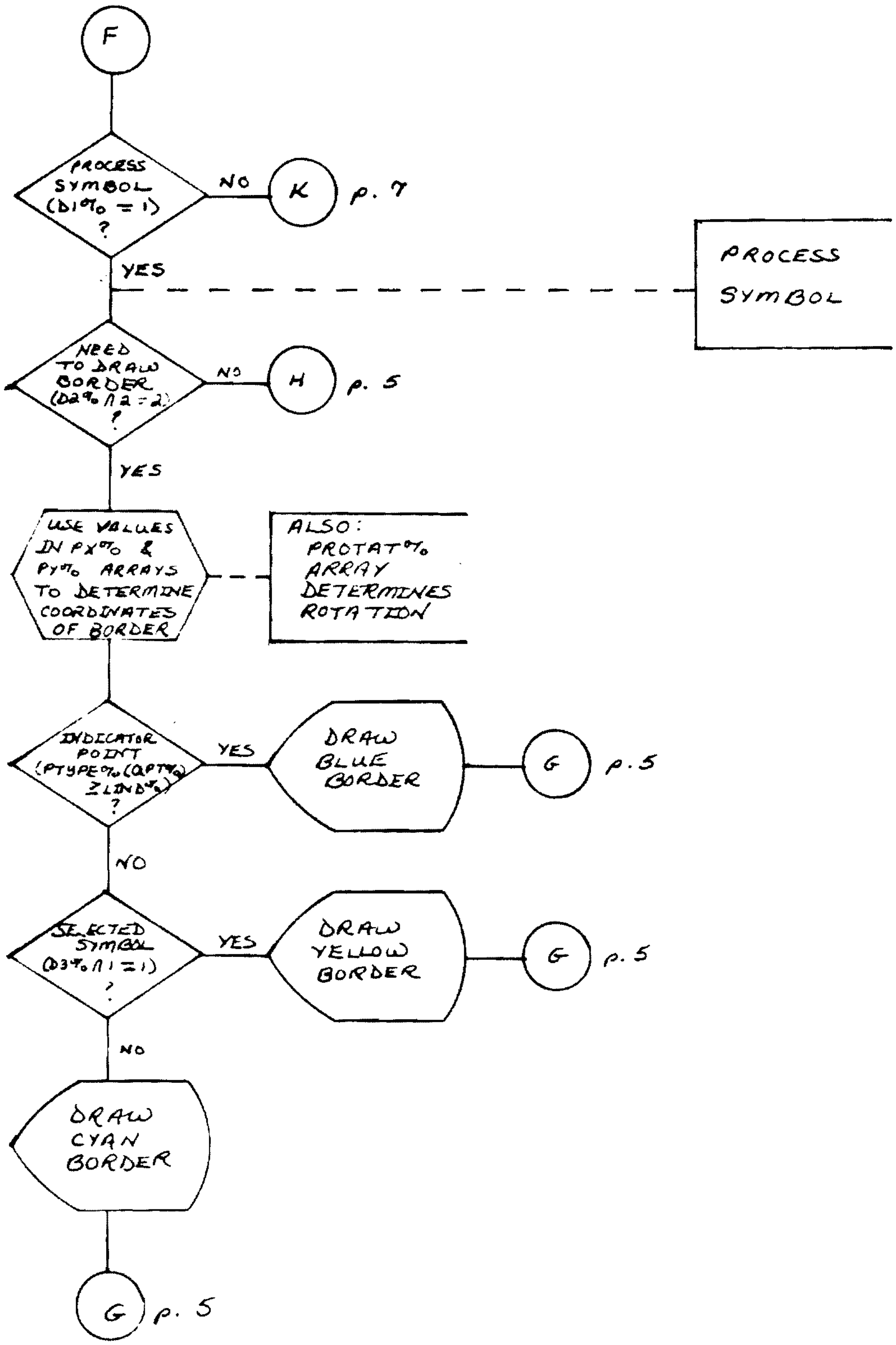
DESIGN NOTES:

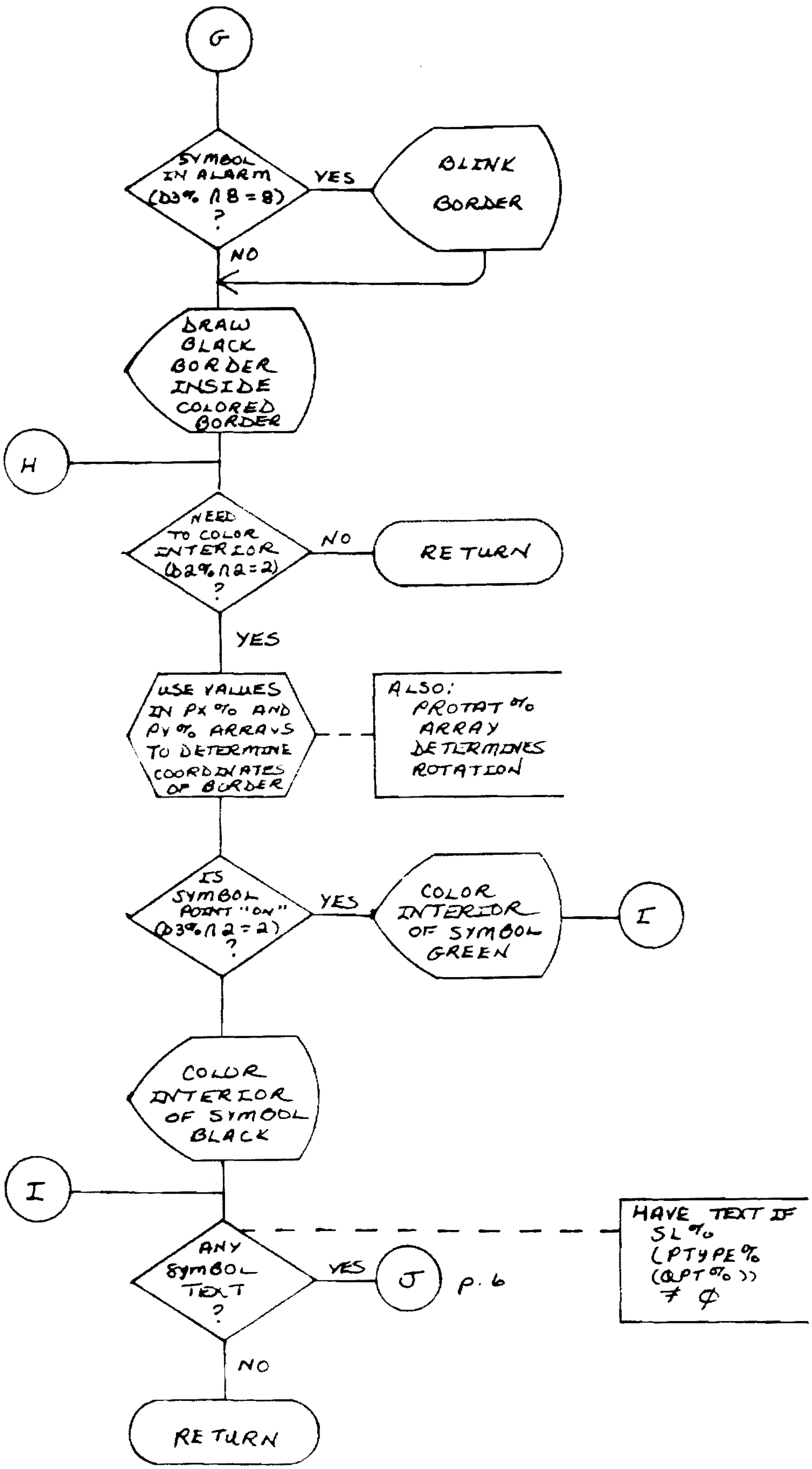
(none)

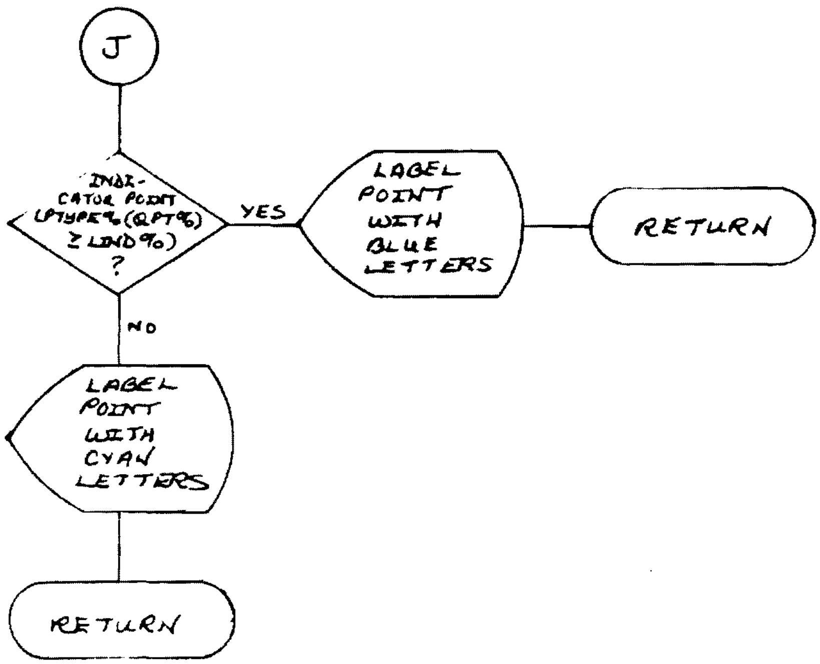

DRAWIT
START
MENU BOX (D1% = -1) ?
NO
B
p. 2
YES
MENU ITEM
SELECTED ITEM (D3% ∧ 1 = 1) ?
YES
USE YELLOW
A
NO
USE CYAN
A
ALARM (D3% ∧ 8 = 8) ?
YES
USE BLINKING MAGENTA
NO
GET Y-COORDINATE OF BOX FROM BOXLOC%(@MENU%)
@MENU% = CURRENT MENU ITEM
GET TEXT FOR ITEM FROM TT$(DIMEN%(@MENU%))
DIMEN% = CURRENT MENU
SET POINT LIMITS (@FCN% = 4) ?
YES
DISPLAY LTMP, HTMP, OR STMP (BASED ON @MENU%)
RETURN
NO
RETURN

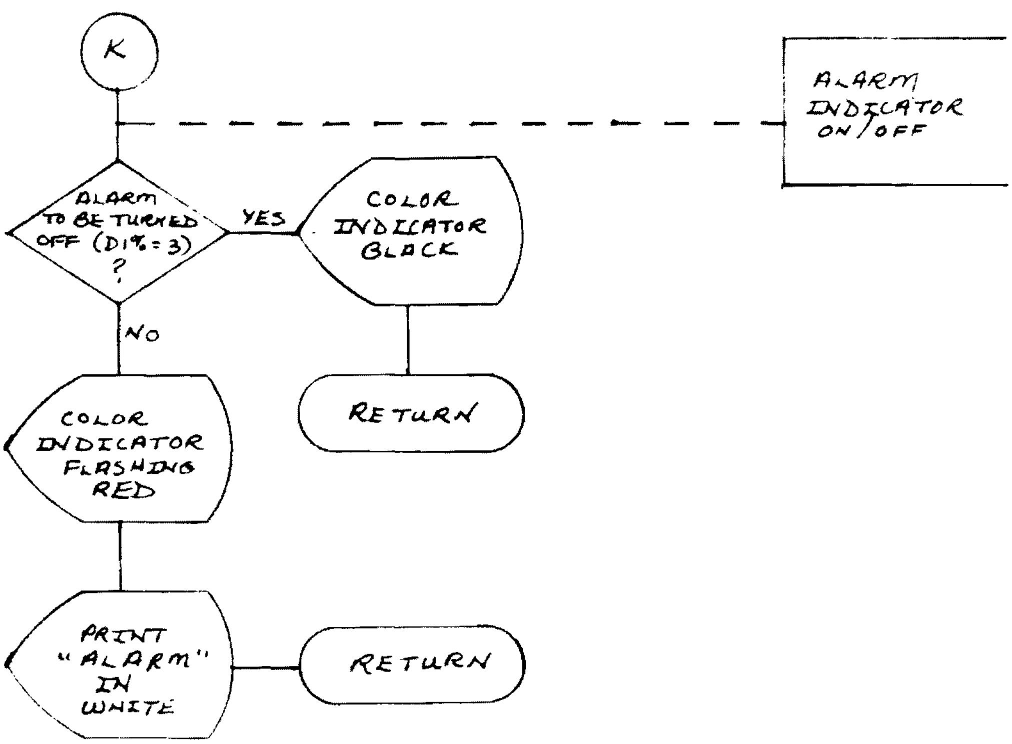

B
FUNCTION KEY (D1% = Ø) ?
NO
F
p. 4
YES
FUNCTION KEY
GET Y-COORDINATE OF KEY FROM
SELECTED KEY (D3%(1) = 1) ?
YES
COLOR KEY YELLOW
C
NO
COLOR KEY PER KCOLR% (1, @FCN%)
C
GET KEY TEXT COLOR FROM KCOLR% (2, @FCN%)
SELECTED KEY (D3%(1) = 1) ?
YES
USE BLACK TEXT ON YELLOW
NO
AUTO/MANUAL KEY (@FCN% = 5) ?
YES
D
p. 3
E
p. 3


D
IS A DE DIS-PLAYED (GDTYPE% = 1) ?
NO
E
YES
DE IN AUTO MODE (FDEAM% = 1) ?
YES
PRINT "AUTO" IN BLUE
RETURN
NO
PRINT "MANUAL" IN RED
RETURN
E
PRINT LINES OF KEY TEXT (IN KTEXT$ (i, @FCN%))
i = 1, 3
RETURN


F
PROCESS SYMBOL (D1% = 1) ?
NO
K
p. 7
YES
PROCESS SYMBOL
NEED TO DRAW BORDER (D2% ∩ 2 = 2) ?
NO
H
p. 5
YES
USE VALUES IN PX% & PY% ARRAYS TO DETERMINE COORDINATES OF BORDER
ALSO: PROTAT% ARRAY DETERMINES ROTATION
INDICATOR POINT (PTYPE%(QPT%) ≥ LIND%) ?
YES
DRAW BLUE BORDER
G
p. 5
NO
SELECTED SYMBOL (D3% ∩ 1 = 1) ?
YES
DRAW YELLOW BORDER
G
p. 5
NO
DRAW CYAN BORDER
G
p. 5


G
SYMBOL IN ALARM (D3% ∩ 8 = 8) ?
YES
BLINK BORDER
NO
DRAW BLACK BORDER INSIDE COLORED BORDER
H
NEED TO COLOR INTERIOR (D2% ∩ 2 = 2) ?
NO
RETURN
YES
USE VALUES IN PX% AND PY% ARRAYS TO DETERMINE COORDINATES OF BORDER
ALSO: PROTAT% ARRAY DETERMINES ROTATION
IS SYMBOL POINT "ON" (D3% ∩ 2 = 2) ?
YES
COLOR INTERIOR OF SYMBOL GREEN
I
COLOR INTERIOR OF SYMBOL BLACK
I
ANY SYMBOL TEXT ?
HAVE TEXT IF SL% (PTYPE% (QPT%)) ≠ Φ
YES
J
p. 6
NO
RETURN


J
INDI-CATOR POINT (PTYPE%(QPT%) ≥ LIND%) ?
YES
LABEL POINT WITH BLUE LETTERS
RETURN
NO
LABEL POINT WITH CYAN LETTERS
RETURN


K
ALARM INDICATOR ON/OFF
ALARM TO BE TURNED OFF (D1% = 3) ?
YES
COLOR INDICATOR BLACK
RETURN
NO
COLOR INDICATOR FLASHING RED
PRINT "ALARM" IN WHITE
RETURN

NAME: ADISABLE

PURPOSE:

The ADISABLE module is used to disable the current alarm point for a period of 5 minutes.

OPERATIONAL DESCRIPTION:

The ADISABLE module makes use of the GJTS and IJTS subroutine to compute the Julian time at which the disablement period will expire. It then stores the alarm point identifier along with the computed time in the alarm disablement queue. DRAWIT is called to stop the alarm points symbol from flashing and to turn off the ealarm indicator.

CALLED BY:

XSD

XSE

XSPL

CALLS:

DISMESS

DRAWIT

GJTS

IJTS

NAME: ADISABLE (continued)

PASSED ARGUMENTS:

AQUE#	- Alarm disablement queue
APT%	- Current alarm point
BQUE%	- Bottom of queue pointer plus 1
CDBVL	- Current database value for points
DSTAT%	- Disable status flag for points
IJT#	- Incremented Julian Time
MQUE%	- Maximum number of alarms in alarm queue
QPT%	- Currently selected point
TQUE%	- Top of queue pointer

RETURNED ARGUMENTS:

APT%	- Current alarm point
AQUE#	- Alarm disablement queue
BQUE%	- Bottom of queue pointer plus 1
DSTAT%	- Disable status flag for points
SALARM%	- Flag to determine whether or not to sound alarm

FILE INPUT/OUTPUT:

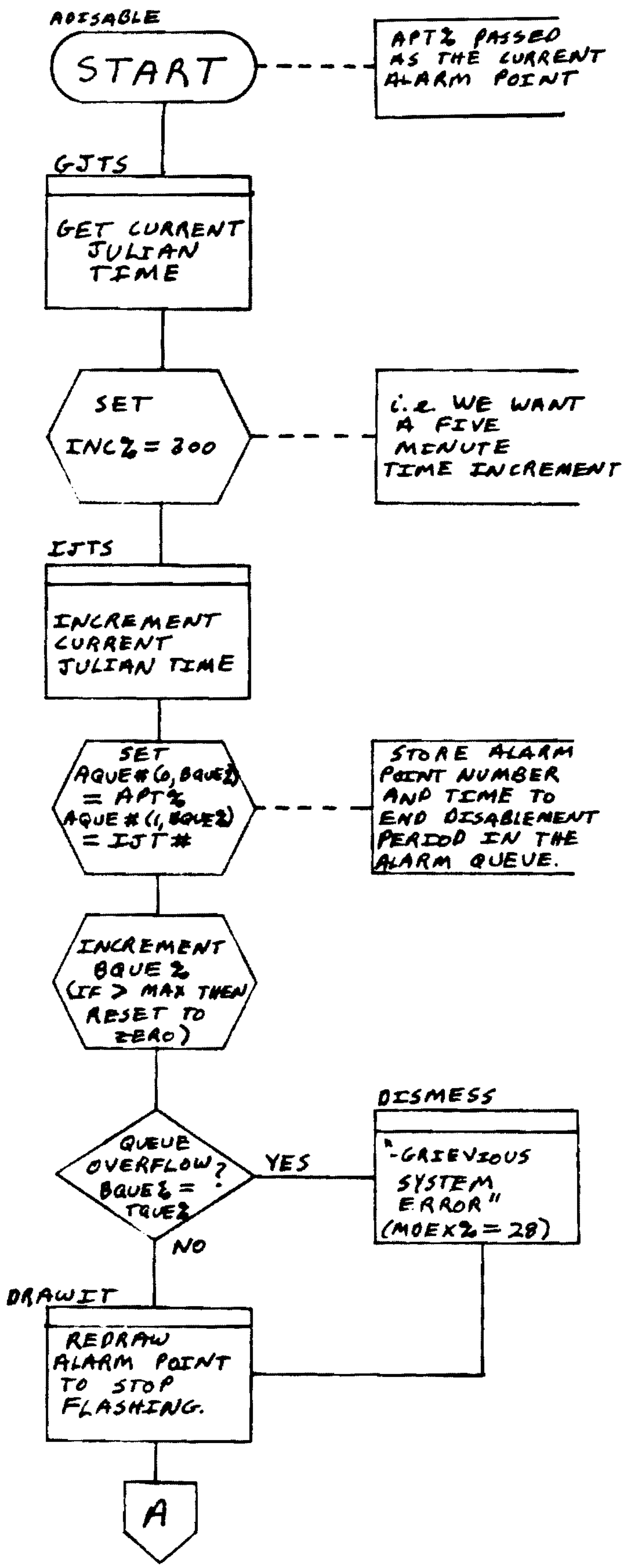
(none)

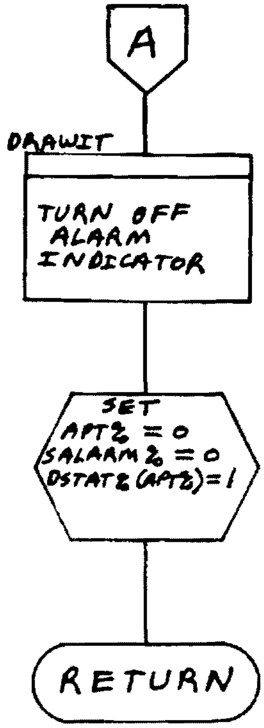
HARDWARE INTERACTION:

CRT Screen - Displays messages in text area

DESIGN NOTES:

(none)


ADISABLE
START
APT% PASSED AS THE CURRENT ALARM POINT
GJTS
GET CURRENT JULIAN TIME
SET INC% = 300
i.e. WE WANT A FIVE MINUTE TIME INCREMENT
IJTS
INCREMENT CURRENT JULIAN TIME
SET AQUE#(0,BQUE%) = APT% AQUE#(1,BQUE%) = IJT#
STORE ALARM POINT NUMBER AND TIME TO END DISABLEMENT PERIOD IN THE ALARM QUEUE.
INCREMENT BQUE% (IF > MAX THEN RESET TO ZERO)
QUEUE OVERFLOW? BQUE% = TQUE%
YES
NO
DISMESS
"GRIEVIOUS SYSTEM ERROR" (MOEX% = 28)
DRAWIT
REDRAW ALARM POINT TO STOP FLASHING.
A


A
DRAWIT
TURN OFF ALARM INDICATOR
SET
APT% = 0
SALARM% = 0
DSTAT%(APT%)=1
RETURN

NAME: GJTS

PURPOSE:

The GJTS subroutine is used to obtain the current Julian time/date.

OPERATIONAL DESCRIPTION:

The GJTS subroutine obtains values for the current time from the memory locations maintained by the assembly level clock subroutine. It uses this information, together with the last computed Julian time, to obtain a new value for the current Julian time, CJT#.

CALLED BY:

ADISABLE
BUZOFF
MMI EXEC
UPDATE

CALLS:

DISMESS

PASSED ARGUMENTS:

CJT#	- Current Julian Time
JDAY%	- Julian day matrix
D%	- Numerical value of day in the year (date)
M1%	- Leap year indicator flag
Y%	- Numerical value of year

Values in locations 4321-4324 Hex

NAME: GJTS (continued)

RETURNED ARGUMENTS:

Memory Location 4234 Hex

CJT#	-	Current Julian Time
D%	-	Numerical value of date
H%	-	Numerical value of hour
M%	-	Numerical value of minutes
M2%	-	Numerical value of day of the month
MT%	-	Numerical value of the month
S%	-	Numerical value of the seconds
Y%	-	Numerical value of year

FILE INPUT/OUTPUT:

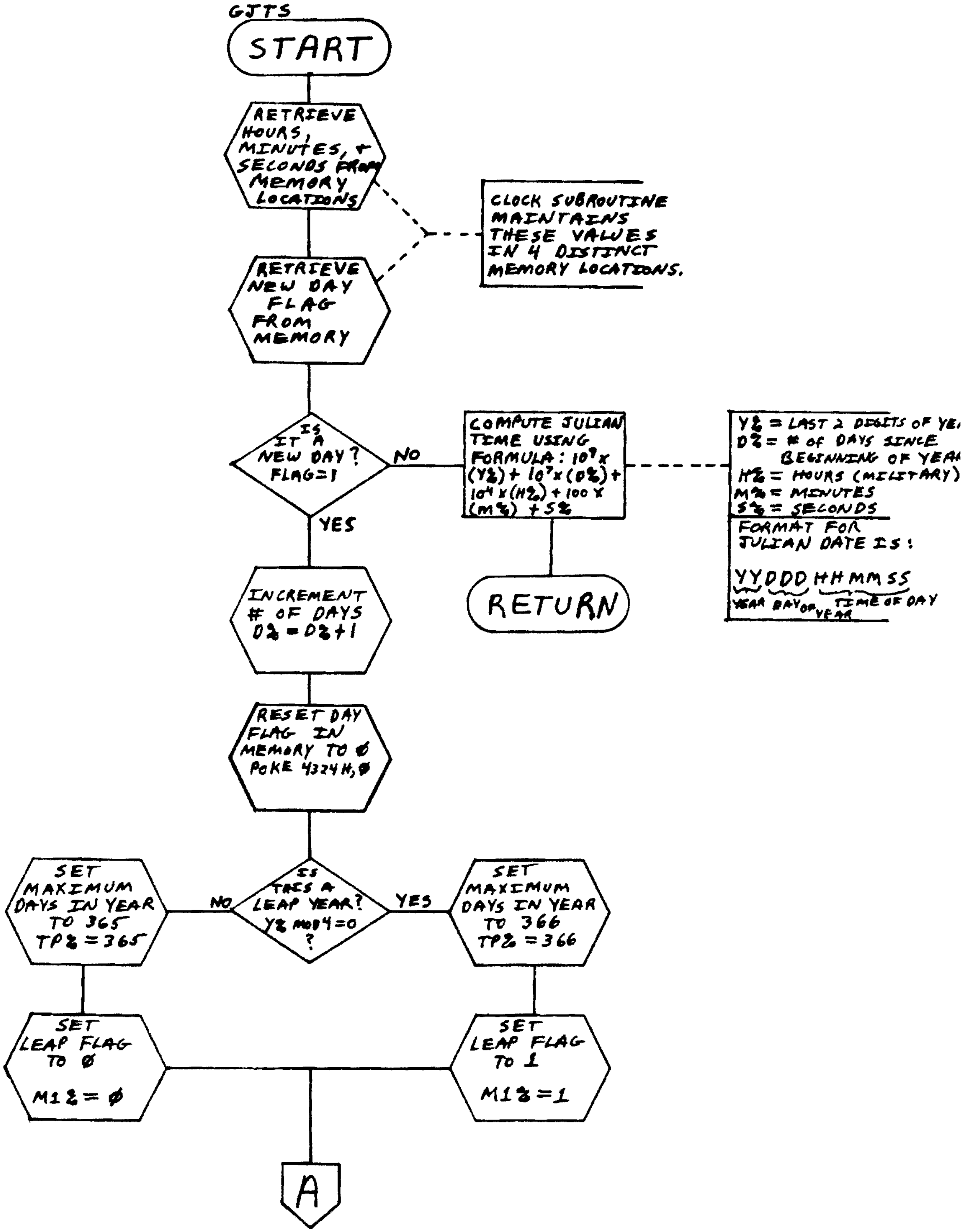
(none)

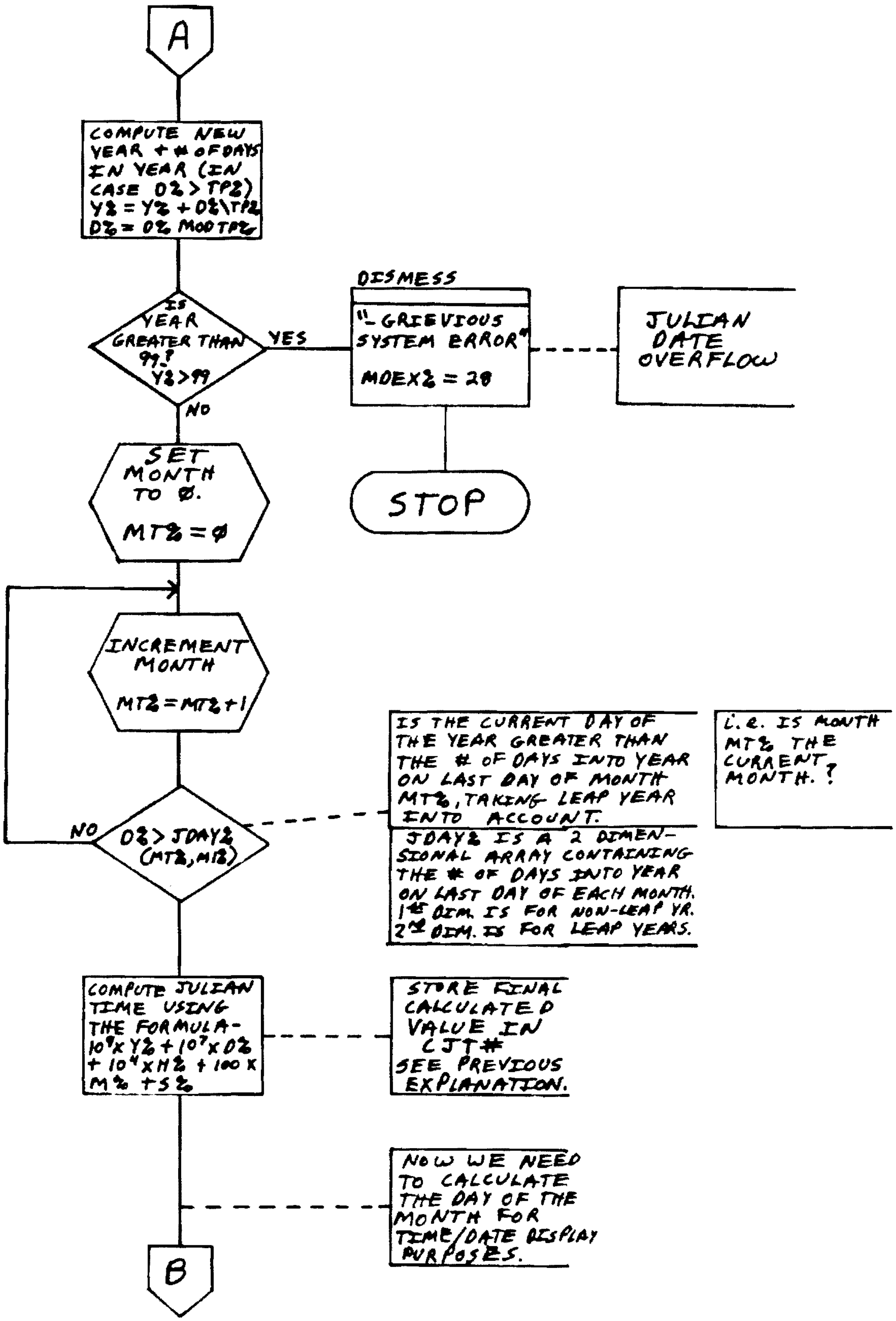
HARDWARE INTERACTION:

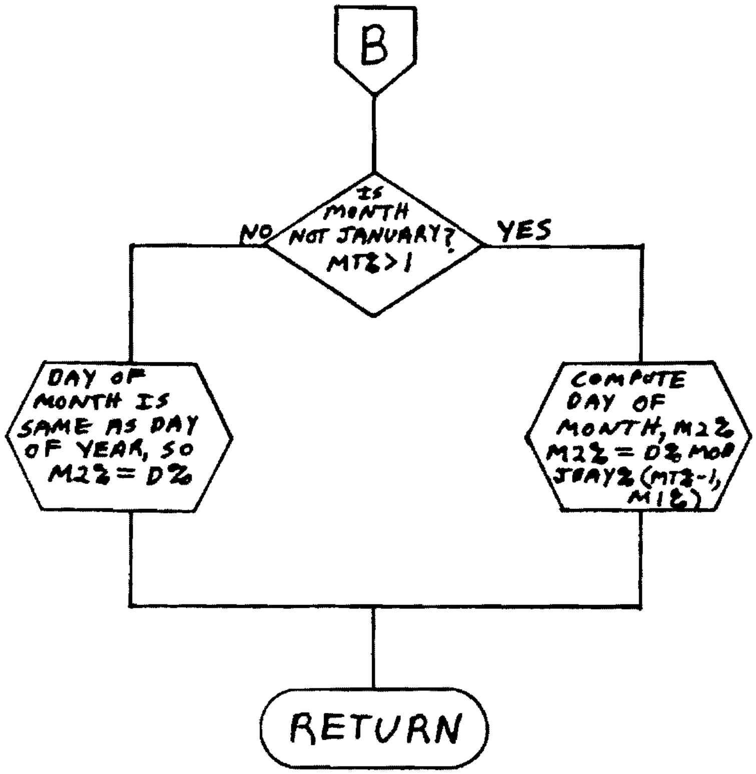
Probes memory directly

DESIGN NOTES:

Julian time is used to provide an absolute standard time reference for all time related functions. The format for the variable CJT# is YYDDDHHMMSS.


GJTS
START
RETRIEVE HOURS, MINUTES, + SECONDS FROM MEMORY LOCATIONS
RETRIEVE NEW DAY FLAG FROM MEMORY
CLOCK SUBROUTINE MAINTAINS THESE VALUES IN 4 DISTINCT MEMORY LOCATIONS.
IS IT A NEW DAY? FLAG=1
NO
YES
COMPUTE JULIAN TIME USING FORMULA: 10⁹ x (Y%) + 10⁷ x (D%) + 10⁴ x (H%) + 100 x (M%) + S%
Y% = LAST 2 DIGITS OF YEA
D% = # of DAYS SINCE BEGINNING OF YEA
H% = HOURS (MILITARY)
M% = MINUTES
S% = SECONDS
FORMAT FOR JULIAN DATE IS:
YYDDDHHMMSS
YEAR DAY OF YEAR TIME OF DAY
RETURN
INCREMENT # OF DAYS D% = D% + 1
RESET DAY FLAG IN MEMORY TO Ø POKE 4324H,Ø
IS THIS A LEAP YEAR? Y% MOD 4 = 0 ?
NO
YES
SET MAXIMUM DAYS IN YEAR TO 365 TP% = 365
SET MAXIMUM DAYS IN YEAR TO 366 TP% = 366
SET LEAP FLAG TO Ø M1% = Ø
SET LEAP FLAG TO 1 M1% = 1
A


A
COMPUTE NEW YEAR + # OF DAYS IN YEAR (IN CASE D% > TP%) Y% = Y% + D%\TP% D% = D% MOD TP%
IS YEAR GREATER THAN 99? Y%>99
YES
NO
DISMESS
"_GRIEVIOUS SYSTEM ERROR" MOEX% = 28
JULIAN DATE OVERFLOW
STOP
SET MONTH TO Ø. MT% = Ø
INCREMENT MONTH MT% = MT% + 1
NO
D% > JDAY% (MT%, M1%)
IS THE CURRENT DAY OF THE YEAR GREATER THAN THE # OF DAYS INTO YEAR ON LAST DAY OF MONTH MT%, TAKING LEAP YEAR INTO ACCOUNT.
i.e. IS MONTH MT% THE CURRENT MONTH?
JDAY% IS A 2 DIMENSIONAL ARRAY CONTAINING THE # OF DAYS INTO YEAR ON LAST DAY OF EACH MONTH. 1st DIM. IS FOR NON-LEAP YR. 2nd DIM. IS FOR LEAP YEARS.
COMPUTE JULIAN TIME USING THE FORMULA- 10^9 X Y% + 10^7 X D% + 10^4 X H% + 100 X M% + S%
STORE FINAL CALCULATED VALUE IN CJT# SEE PREVIOUS EXPLANATION.
NOW WE NEED TO CALCULATE THE DAY OF THE MONTH FOR TIME/DATE DISPLAY PURPOSES.
B


B
IS MONTH NOT JANUARY? MT%>1
NO
YES
DAY OF MONTH IS SAME AS DAY OF YEAR, SO M2% = D%
COMPUTE DAY OF MONTH, M2%
M2% = D% MOD JDAY%(MT%-1, M1%)
RETURN

NAME: IJTS

PURPOSE:

The IJTS subroutine is used to increment the current Julian time stored in CJT#, by a fixed number of seconds.

OPERATIONAL DESCRIPTION:

The IJTS subroutine uses the time stored in CJT# and computes a new time, IJT#, which is equivalent to CJT# plus INC% seconds.

CALLED BY:

ADISABLE

BUZOFF

MMI EXEC

UPDATE

CALLS:

DISMESS

PASSED ARGUMENTS:

CJT#	- Current Julian Time
D%	- Numerical value of date
H%	- Numerical value of hour
INC%	- Number of seconds to increment Julian time
JDAY%	- Julian day matrix
M%	- Numerical value of minutes
M1%	- Leap year indicator flag
S%	- Numerical value of seconds
Y%	- Numerical value of year

NAME: IJTS (continued)

RETURNED ARGUMENTS:

IJT# - Incremented Julian Time

FILE INPUT/OUTPUT:

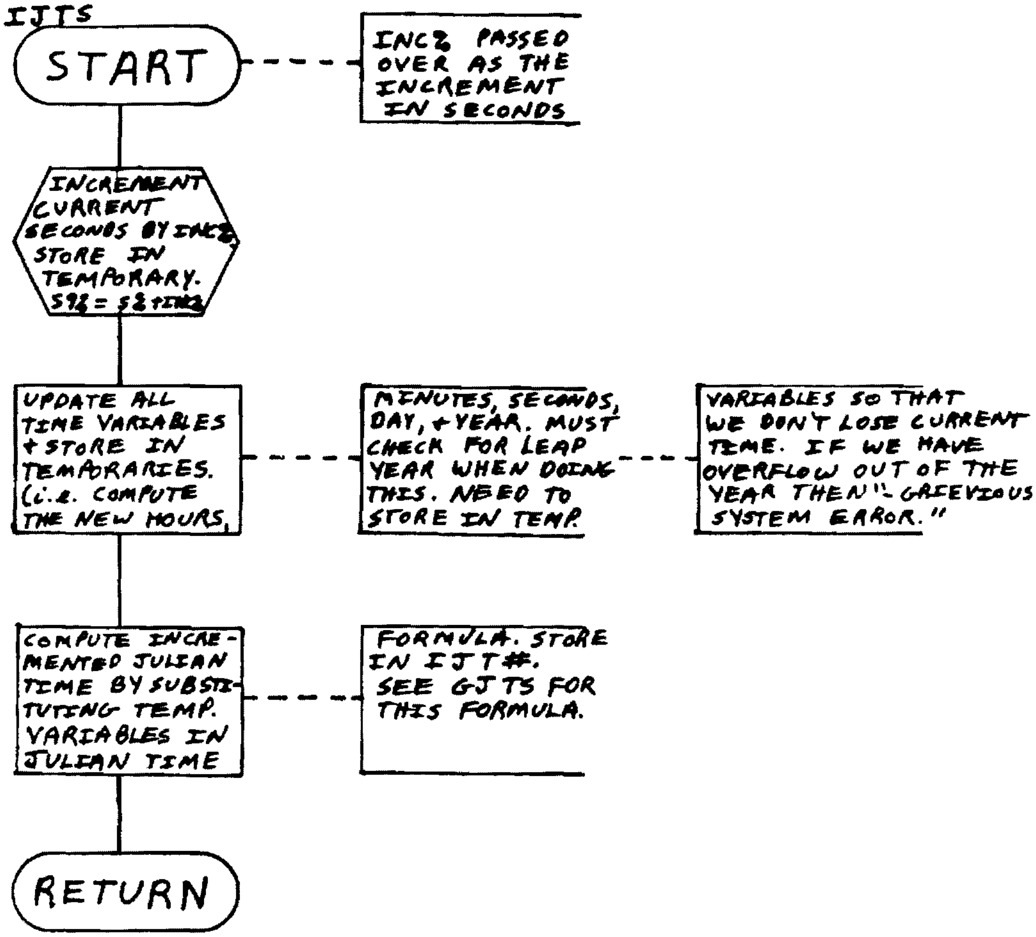
(none)

HARDWARE INTERACTION:

(none)

DESIGN NOTES:

(none)


IJTS
START
INC% PASSED OVER AS THE INCREMENT IN SECONDS
INCREMENT CURRENT SECONDS BY INC%, STORE IN TEMPORARY. S9% = S%+INC%
UPDATE ALL TIME VARIABLES + STORE IN TEMPORARIES. (i.e. COMPUTE THE NEW HOURS,
MINUTES, SECONDS, DAY, + YEAR. MUST CHECK FOR LEAP YEAR WHEN DOING THIS. NEED TO STORE IN TEMP
VARIABLES SO THAT WE DON'T LOSE CURRENT TIME. IF WE HAVE OVERFLOW OUT OF THE YEAR THEN "GRIEVIOUS SYSTEM ERROR."
COMPUTE INCREMENTED JULIAN TIME BY SUBSTITUTING TEMP. VARIABLES IN JULIAN TIME
FORMULA. STORE IN IJT#. SEE GJTS FOR THIS FORMULA.
RETURN

3.3 Implementation Notes

This section describes some of the problems encountered in adapting the MMI design to the Chromatics computer system environment and attempt to clarify why certain implementation decisions were made. It should be noted that many of these problems would not exist in a larger or better developed computer system. The most difficult problem to overcome was, by far, the lack of available central memory.

3.3.1 Composition of Code

The MMI demonstration system has been implemented on a Chromatics CG 3999 color-graphics computer utilizing the CG-BASIC interpreted language. Because of the inherent limitations of this computer system, it was necessary to compact the original source code into a dense form and further to split the system into three program segments. Compaction of the code essentially involves removal of all extra blank spaces, removal of all internal comment statements (annotation), reduction of variable name lengths, and frequent use of multiple statements per line. Such compaction makes it very difficult to read the source code listing, therefore, in order to facilitate understanding of the source code listing, an annotated version of the listing has been included in Appendix E of this manual. An attempt has been made to expand the annotated listing by adding back extra blank spaces and restoring the variable names to their original lengths. In a larger computer system these types of problems would not exist.

3.3.2 Size Considerations

The total MMI program size is between 40K and 42K bytes. The database is approximately 7K bytes, including string storage. The master message file (MESSAGE.DAT) is approximately 5K bytes. Since there are only 28K bytes of effectively usable memory in the CG computer, the program had to be split into three program segments. The program segments have been designed to operate under CG-BASIC with at least 24,311 bytes of available storage. This amount corresponds to typing in &HB070 after the "MEMORY Size?" querry upon initial start-up of the CG-BASIC interpreter. This memory size factor permits the computer to have a 4K byte Create Buffer. Th Create Buffer is needed to display buffer files containing the static graphic pictures used by the software. The three program segments are:

A. Main program segment, MMIMAIN
B. Secondary program segment, MMIPART2
C. Initialization program segment, MMIINIT

3.3.3 Software Delineation (Program Segmentation)

The division of the software into three program segments was accomplished through the logical separation of the function key operations. Those modules supporting function keys which are frequently used and which perform some active process on a data environment were placed in the main program segment (MMIMAIN). Those modules supporting function keys which perform some less frequently used support procedures were placed in the secondary program segment (MMIPART2). The remaining modules were placed in the initialization program segment (MMIINIT) since they pertain to initial system start-up and thus are called only once during each operating session. Enough support software was then added to each program segment to allow it to be as independent as possible given the imposed memory constraints. It was necessary, in some cases, to duplicate part or all of certain modules in more than one segment and to split apart other modules between segments in order to make the system operate. In a larger computer these problems would not exist.

3.3.4 Transfer of Control

The initialization program segment is executed only once. Thereafter, program control is transferred between the main and secondary program segments based upon operator function key selection. Transfer of control is accomplished through the use of the DOS transient command "CHAIN." This command does not affect stored variable values and therefore prevents the need to save and restore the system database each time control is transferred. Several special control variables are maintained which provide for smooth and logical transfer of control between segments. The program segments were developed to reduce the number of control transfers. This was accomplished through the application of the principle of locality (see 3.3.3 above). As such, control of the MMI should remain with the main program segment for the majority of the time.

The following function keys are supported by the main program segment:

A. START/ENABLE
B. STOP/DISABLE
C. DISPLAY DIAGRAM
D. SET POINT/LIMITS
E. AUTO/MANUAL
F. CONFIRM
G. CANCEL

The following function keys are supported by the secondary program segment:

A. PRINT REPORT
B. CHANGE OPER
C. MODIFY SCHED
D. CONFIRM
E. CANCEL

3.3.5 Touch Panel

Since CG-BASIC can not handle touch panel interrupts adequately with a program of this size and complexity, a special touch panel I/O driver was developed. The touch panel I/O driver is merely an assembly language level interrupt service routine designed to handle interrupts from Serial Port # 1. (This port is where the touch panel hardware connection is made.) It executes much faster than interpreted CG-BASIC code and thus can trap the touch panel input in a timely fashion, store it in reserved memory locations, and notify the CG-BASIC MMI program thru the use of a semaphore (a dedicated memory location used for inter-process communication). See Appendix C-2 for further information on the touch panel I/O driver.

3.3.6 Real-Time Clock

The basic CG-Computer had to be equipped with a special count-down timer in order to allow it to simulate a real-time environment. This timer is used as a substitute for the real-time system clock found in most larger computer systems. As such, a special assembly language level program had to be developed in order to operate the timer and interface it to the CG-BASIC MMI program. The real-time clock programs are documented in Appendix C-1.

3.3.7 Assembly-language Level Support Software

It was found necessary to place the Real-time Clock (RTC) initialization routine, RTC Interrupt Service Routine, and the Touch Panel I/O Dirver in the function key portion of the computer's memory. This is due to the fact that the transient program area was exhausted and that this was the only area of memory which could be protected against change. These routines are assembly level programs which run as separate processes on an interruptable basis. It is interesting to note that the CG-Computer is essentially a single process machine. Semi-Multiprocess capability was achieved through the use of interrupts. Thus, the execution of these three support programs occupies only a small fraction of the CG's Z-80 processor time. The rest of the processing time is devoted to the CG-BASIC interpreter.

3.3.8 Touch Panel Calibration

The CG-computer provides a screen resolution of 512 x 512 pixels with its origin located at the lower left corner of the screen. The Carroll touch panel provides a resolution of only 111 x 95 beams and its origin is located in the upper left corner. In order to allow the MMI software to correctly convert touch panel coordinates into screen coordinates, a special mapping function was developed. This function uses a set of eight calibration factors obtained through the use of a special touch panel calibration program. The touch panel calibration program utilizes a third order polynomial curve fitting routine to compute these calibration factors. A third order correction was used because an earlier second order correction proved to be unsuccessful. The third order correction achieves a much more reliable coordinate conversion whose degree of precision proved to be satisfactory for this system.

3.3.9 Master Message File

The MMI software utilizes a special master message file (MESSAGE.DAT) from which it retrieves most of the color-coded messages that are displayed in the text area. All stereotyped messages which did not require a "fill-in-the - blank" type response by the system were placed in this file along with the special mode code sequences necessary to implement the desired color pattern on the screen.

3.3.10 Echoing of Passwords

During the change operator command, the password is echoed on the screen. In a real system this is not a desirable trait; however, in this demonstration system it was necessary in order to insure accurate entry of the password. This is due to the fact that when CG-BASIC performs garbage collection and when a real-time update occurs, the keyboard interrupt service routine is unable to trap any keyboard input. Thus, if the password is not echoed and garbage collection or an update occurs, the operator will unknowingly type in data which the system will be unable to accept.

3.3.11 Sequencing vs. Performance

The sequencing of variable names in the initialization module is very important. Equally important is the sequencing of modules within program segments and the sequencing of files on the disk. This is due to the primitive sequential search algorithm employed by the CG Computer. Critical variable names and sections of code were placed earlier in the code so as to reduce this search time and thus increase system execution speed. Similiarly, critical and frequently used disk files were placed close to the center disk track. Note that any changes in the sequencing of initial variable names, modules, or disk files may result in poor system performance, and possibly, even system failure. Especially critical is the placement of the keyboard interrupt service routine and its associated variables.

3.3.12 CP/M and MicroSoft BASIC

An attempt was made to increase the CG-Computer's transient program area capacity through the use of CP/M to provide for bank memory switching and an effective gain of 32K bytes of memory storage. To this end, a Microsoft BASIC compiler was purchased with the intent of reducing the MMI program size and increasing execution speed. Unfortunately, a satisfactory way was not found to implement the CG graphics and interrupt capability in the MicroSoft BASIC language. Thus, this attempt was abandoned. As a result, the demonstration operates much more slowly.

3.3.13 Correcting Typing Mistakes

From time to time the operator will be requested to type in a number or name. In so doing, it is not uncommon for a typing mistake to be made. The operator can correct such mistakes before the RETURN key is depressed by key (--) or by holding down the control key (CNTRL) while pressing the H key. Both methods perform the exact same function. When a character on the screen is backspaced over, it will be erased and a new character may be typed in the place or the backspace key may be used to erase another character. When the mistake has been corrected the RETURN key should be pressed to tell the computer to accept the new value or name.

MAN-MACHINE INTERFACE DEVICE

APPENDICES TO SOFTWARE DESIGN MANUAL

Man-Machine Interface Device

Appendices to

Software Design Manual

by

R.C. Coleburn

B. S. Rice

J.S. Cathcart

B. B. Wise

of the

Command and Control Branch
Electronics and Computer Systems Laboratory
Engineering Experiment Station
Georgia Institute of Technology
Atlanta, Georgia 30332

GIT/EES Project A-3008

May 1982

Prepared for

NAVAL CIVIL ENGINEERING LABORATORY
Port Hueneme, California 93043

TABLE OF CONTENTS

APPENDIX A

EQUIPMENT LIST

APPENDIX A

EQUIPMENT LIST

This apppendix contains a listing of the hardware used to implement the MMI demonstration device, along with available specifications for each item.

Color Graphics Computer - Chromatics Model CG3999

GENERAL

Power

105-125 volts, 50/60 Hz, 600 watts.

TEMPERATURE

+10'C to +40'C Operating.

-30'C to +70'C Storage.

HUMIDITY

0-95%, noncondensing.

PACKAGE COLOR

Light Beige (Federal Standard 26521)

and Brown (Federal Standard 20140).

X-RADIATION

Less than 0.5 milliroentgen/hour at a distance of 2 inches from all exterior surfaces.

Color Graphics Computer - Chromatics Model CG3999

PROCESSOR

Z-80 CPU

REFRESH MEMORY

HIGH Resolution Model - 131,072 bytes dynamic RAM

PROGRAM MEMORY

8,192 bytes of EPROM and 1,024 bytes of STATIC RAM for base routines and CRT Operating System.

SOFTWARE

PROCESSORS

CRT Operating System with full, local, and half duplex modes as well as ESCAPE code processing and device assignment capability.

INTERFACES

SERIAL

Serial I/O port - asynchronous; independently programmable from 110 to 31,250 baud (9,600 baud highest standard rate); 1.5 stop bits, programmable to 1 or 2; TTL and RS-232C interface with busy output lines (clear-to-send status line also included in RS-232C interface).

DISPLAY FUNCTIONS

GRAPHIC MODE

DOT, Incremental DOT, X BAR, Incremental X BAR, Y BAR, Incremental Y BAR, Vector, Concatenated Vector.

ALPHANUMERIC MODE

ASCII and special characters available from RAM or EPROM positioned to any dot position on screen.

COORDINATE ENTRY

Relative and absolute modes of decimal digits, binary codes, or cursor position.

CHARACTER FORMAT

96 ASCII upper and lower case 5x7 dot matrix characters, and 96 special 6x10 dot matrix graphics characters programmed in EPROM memory.

CHARACTER MAGNIFICATION

Individually selectable in X or Y to any integer multiplier.

CURSOR

Programmable in color, position, and visibility (on or off).

CHARACTER INTERLINE SPACING

Variable, up to 255 raster lines.

CHARACTER WRITE DIRECTION

Vertical or horizontal.

CHARACTER RESOLUTION

512 x 512 dot resolution - 85 characters/line by 51 lines/page.

CONTROL FUNCTIONS

Cursor Up, Down, Left and Right (character and dot spacing), Erase Page, Erase Line, Home, Tab, Carriage Return, Line Feed, and Backspace.

WINDOWS

4 each, individually programmable in size, position, and all the previously described functions including a separate cursor for each window. Each window is a physical output device and may receive data from any physical input device or logical output device when properly assigned.

VIDEO DISPLAY

Diagonal Measure		PHOSPHOR SIZE		Phosphor Area	
(IN)	(CM)	(IN)	(CM)	(IN^2)	(CM^2)
19	48.3	11.70x15.61	29.7x39.6	182.67	1178

SCREEN SIZE

USABLE DISPLAY AREA

(IN)	(CM)	(IN^2)	(CM^2)	FORMAT
10.25x14.50	26.0x36.8	148.63	958.9	512x512

REFRESH RATE

60 times/second non-interlace, synchronized to 60 Hz line frequency or 50 times/second non-interlaced, synchronized to 50 Hz line frequency.

COLOR LEVELS

8 foreground and 8 background - red, green, blue, magenta, cyan, yellow, white, and black.

CONVERGENCE

9 sector, with each sector individually converged from front drawer accessible controls.

CONTROLS

Brightness, focus, convergence, on/off.

DEFLECTION

Magnetic.

Color Graphics Computer - Chromatics Model CG3999 (continued)

FOCUS

High voltage electrostatic.

BLINK RATE

1.9 Hz, cursor and dots, foreground and/or background.

CURSOR

4 each, 1 per window, programmable in position, color and visibility (on or off). Expands vertically to match character height when character Y magnification is not equal to 1.

Options added to the basic CG3999:

Option 15 - Expanded card cage
Option 17 - Blink
Option 83 - Keyboard
Option 23 - Memory card w/32k bytes of RAM
Option 31 - Second serial I/O port
Option 41 - Floppy disk controller w/DOS software
Option 43 - Dual standard floppy disk
Option 61 - CPU operating system
Option 62 - Text editor
Option 63 - Z-80 assembler
Option 64 - BASIC language interpreter
Option 76 - Complex boundary fill/pattern
Option 77 - Extended graphics/alphanumerics/zoom
Option 56 - Light Pen
Option 39 - GPIO
Option 38 - 3rd and 4th serial port
Option 65 - Function key processor
Option 54 - Programmable timer

Touch Panel - Carroll 19" IR Touch Input Kit:

CRT Display Size		19"
Matrix		
	X Axis	56
	Y Axis	48
Resolution		
	X Axis	.275
	Y Axis	.250
Touch Active Area		
	X Axis	15.125
	Y Axis	11.750

Scan Rate 40 - 60 CPS depending on size

Data Format — Four ASCII characters are generated per HIT

1. Uncover character (identifies Touch Data)
2. X data character
3. Y data character
4. Stop code

Baud Rate — Selectable on power box (150, 300, 600, 1200, 2400, 4800, 9600)

Touch Panel - Carroll 19" IR Touch Input Kit (continued)

Touch Input Features

- All Solid State construction
- Microprocessor Controller
- No obstruction of viewing area
- High reliability
- RS232 interface

Output Printer - NEC Model 7715 Spinwriter:

PASTE UP

Options added to the basic 7715:

Cut-sheet feeder

Vertical forms tractor

APPENDIX B

CUES, ALARMS, and ERROR MESSAGES

APPENDIX B

CUES, ALARMS, and ERROR MESSAGES

All cue, error, and alarm, messages are printed in color on the screen using the following color code scheme:

- o Cues - printed in green
- o Errors - printed in red
- o Alarms - printed in white on a red blinking background.

This Appendix follows the same color coding convention in that the three types (colors) of messages are broken into separate lists.

CUES - GREEN LIST

Cue Message	Explanation
- Auto Mode Selected For DE MDEX% = 1	If the currently displayed DE is in Manual Mode and the AUTO/MANUAL key is touched, this cue appears to notify the operator that the automatic mode has been requested for this DE. The computer will not execute the mode change command until the CONFIRM ACTION key is touched.
- Command Action Cancelled MDEX% = 2	This cue appears when the operator touches the CANCEL ACTION key while entering a command sequence. The command sequence is aborted and the screen returns to its previous state.
- Command Action Completed MDEX% = 3	After the CONFIRM ACTION key is touched, the system will attempt to execute the command. If it is successful, this message will be given.
- Device (x) Has Been Started/Enabled	This cue signals the operator that the EMCS computer has started/enabled device x, where x is a device which the operator manually commanded the system to start.
- Device (x) Has Been Stopped/ Disabled	This cue signals the operator that the EMCS computer has stopped/disabled device x, where x is a device which the operator manually commanded the system to stop.
- Enter Password From Keyboard MDEX% = 4	This cue appears during the change operator command sequence upon successful acceptance of the operator°s name. At this point, the operator°s password should be typed in using the keyboard.

CUES - GREEN LIST (continued)

Cue Message	Explanation
- Manual Mode Selected For DE MDEX% = 5	If the currently displayed DE is in automatic mode and the AUTO/MANUAL key is touched, this cue appears to notify the operator that the manual mode of operation has been requested for this DE. The computer will not execute the mode change command until the CONFIRM ACTION key is touched.
- Mode Changed On DE (x)	The operator has used the AUTO/MANUAL function to change the mode of operation for DE (x). The system is acknowledging the fact that it has processed the mode change request.
- New Device Selected MDEX% = 6	This cue appears if the operator touches a second device symbol when one has already been selected. The old choice is deselected and the new choice is highlighted.
- New Function Selected MDEX% = 7	This cue appears if the operator touches one of the first eight keys when one has already been selected. The old choice is deselected and the new choice is backlighted.
- New Menu Item Selected MDEX% = 8	This cue appears if the operator touches a second menu item when one has already been selected. The old choice is deselected and the new choice is highlighted.
- New Set Point/Limits Accepted For Point (x)	The operator has used the SET POINT/LIMITS function to change the set point and/or limits for device (x). The system is acknowledging the fact that it has processed the new values and added them to the data base.

CUES - GREEN LIST (continued)

Cue Message	Explanation
- Password Accepted. New Operator Is (x).	If the password matches the one approved for the new operator during the CHANGE OPER command, this cue appears in the text area and the Date/Time/Operator display is updated. X is the new operator°s name.
- Please Type In The New Value MDEX% = 10	This cue appears during the SET POINT/LIMITS command sequence. It instructs the operator to type in a new value for the parameter that is to be modified.
- Ready To Accept Command MDEX% = 11	This cue appears whenever the MMI is idle and waiting for the operator to enter a command sequence. The cue is not imperative, but rather it is simply used to indicate that the MMI computer is operational and ready to be used.
- Ready To Change Operation MDEX% = 12	This cue appears when the operator touches the CHANGE OPER key.
- Ready To Modify Schedule MDEX% = 14	When the MODIFY SCHED command is entered, this cue appears in the text area to inform the operator that the computer is ready to make the desired schedule changes for the current DE.
- Ready To Modify Set Point And/Or Limits MDEX% = 15	When the SET POINT/LIMITS key is touched, this cue appears in the text area to inform the operator that the computer is ready to make the desired analog limit changes to the device selected.

Cue Message	Explanation
- System Shutdown MDEX% = 16	The operator has touched the CHANGE OPER function key and requested a system shutdown. This message indicates that the system has been turned off (stopped, shutdown) by the operator.
- Touch Appropriate Device Symbol MDEX% = 17	When a function key requiring a device selection is touched prior to the device symbol, this cue appears instructing the operator to select the appropriate device.
- Touch CONFIRM ACTION To Execute MDEX% = 18	This cue appears in the text area when all of the parts of a command sequence have been entered. If CONFIRM ACTION is touched, the computer will execute the command.
- Or, Touch CONFIRM ACTION When All Values Appear As Desired MDEX% = 19	This cue appears in the text area when the MODIFY SCHED function is selected. It is used to remind the operator that the computer will not execute any changes to the schedule or to control values until the CONFIRM ACTION key is touched.
- Touch Desired Function MDEX% =20	When a device symbol is touched prior to a function, this cue appears instructing the operator to touch the desired function.
- Touch Square Beside Building/Floor Plan Desired MDEX% =21	When the DISPLAY DIAGRAM key is touched, a menu of available Data Environments appears in the graphics area. This cue instructs the operator to select which DE he wishes to have displayed.

Cue Message	Explanation
- Touch Square Beside Desired Report MDEX% = 22	When the PRINT REPORT key is touched, a menu of available reports appears on the screen. This cue appears in the text area instructing the operator to select the desired report to be printed.
- Touch Square Beside Desired Value To Be Changed MDEX% = 23	The operator has entered either the MODIFY SCHED or SET POINT/LIMITS command. To modify a value, it must be selected. This is done by touching the square beside the desired value.
- Press RETURN To Continue Or S Then RETURN To Stop MDEX% = 24	The operator is examining the HELP routine. This message is a prompt signaling that the system is waiting for a response from the operator before continuing with the next screen of HELP information. If the operator types RETURN, the next screen will be displayed. If the operator types S RETURN, then the HELP sequence will be exited.
-Touch CONFIRM ACTION to Confirm Report Selection MDEX% = 9	After a report is selected from the menu of reports this message appears in the text area. This cue instructs the operator to touch CONFIRM ACTION to have his selection executed.
-Enter New Operator Name From Keyboard MDEX% = 13	After the CHANGE OPER function key is touched and the operator requests to change the system operator, this cue appears in the text area. It is used to inform the operator that the system is ready to accept a new operator.

Cue Message	Explanation
-Touch Square Beside Desired Operation MDEX% = 43	After the CHANGE OPER function key is touched, a menu of available operations appears in the graphics area of the screen. This cue prompts the operator to select an operation.
-Touch Square Beside Desired Report Output Device MDEX% = 47	After the operator has touched the PRINT REPORT function key and requested a Report, the system places a menu of available Report Output Devices on the screen. This cue prompts the system operator to select a device.
-Touch Square Beside Desired Schedule To Be Modified MDEX% = 48	When the MODIFY SCHED function key is touched, a menu of available schedules appears on the screen. This cue appears in the text area instructing the operator to select the desired schedule to be modified.
-Touch CONFIRM ACTION To Confirm Schedule Selection MDEX% = 49	When a Schedule selection has been made by the operator, this cue appears in the text area. The message instructs the operator to touch CONFIRM ACTION to execute the command.
-Touch Desired Value To Be Changed MDEX% = 50	After a schedule has been selected by the operator and displayed on the screen, this message will appear in the text area. This cue instructs the operator to touch a value on the schedule to be modified.
-Press RETURN To Continue... MDEX% = 54	If an operator selects the CRT Screen as the Report Output Device, the Report will appear in the graphics area of the display. This cue instructs the operator to Press the RETURN key on the keyboard after viewing the report in order for system operation to continue.

ERRORS - RED LIST

Error Message	Explanation	Corrective Action
- Command Inappropriate For Device, No Action Taken MDEX%=25	A command sequence cannot be executed because the command is not appropriate for the device. For example, a fan does not have analog values. An attempt to set analog values for the fan will yield this error message.	Enter a new command sequence.
- Device Already Started/Enabled, No Action Taken MDEX% = 26	The operator has attempted to manually stop or disable a device which is already stopped/disabled.	Select a new device or function.
- Device Already Stopped/Disabled, No Action Taken MDEX% = 27	The operator has attempted to manually stop or disable a device which is already stopped/disabled.	Select a new device
- Grievous System Error MDEX% = 28	This message indicates an internal fault of the system. It should never appear under normal circumstances.	Continue processing from point of error.
- High Limit Can Not Be Lower Than Low Limit, Please Reenter MDEX% = 29	The operator has attempted to set the high limit for an analog device lower than the current low limit using the SET POINT/LIMITS command.	Enter a value for the high limit which is greater than the current low limit, or reset the low limit to a value which is less than the new high limit.
- Incomplete Command Sequence, Please Continue MDEX% = 31	The operator has touched the CONFIRM ACTION key before the entire command sequence was entered. For example, the operator requested the STOP function when no device was selected.	The operator may now continue entering the command.

Error Message	Explanation	Corrective Action
- Low Limit Cannot Be Higher Than High Limit, Please Reenter MDEX% = 32	The operator has attempted to set the low limit for an analog device higher than the current high limit using the SET POINT/ LIMITS command.	Enter a value for the low limit which is less than the current high limit or reset the high limit to a value which is greater than the new low limit.
- Name Not Recognized, Access Denied MDEX = 33	The name typed in for the new operator is unknown to the system.	Enter a new command sequence. If the CHANGE OPER function is to be selected again, check to make sure the operator name is typed correctly before typing the carriage return key. If the error occurs again, see the level 2 operator to make certain that the operator name has been placed on the list of approved system operators.
- Password Not Recognized, Access Denied MDEX% = 34	The password entered does not match the one approved for the operator; therefore, access to the system is denied.	Enter a new command sequence. If the CHANGE OPER function is to be selected again, make certain that the password is entered correctly. If the error occurs again, see the level 2 operator to obtain the correct password.
- Set Point Not Within Range Limits, Please Reenter MDEX% = 35	The operator has attempted to change the set point to a value outside of the current range limits for a particular analog device.	Enter a value for the set point that is within the current range. Alternately, change the range.
- Typographical Error, Please Reenter The Data Item. MDEX% = 36	The operator has made a typographical error while entering data from the keyboard. Typical errors include typing a letter within a number or comma within a number.	Correct the error by retyping the correct item.

Error Message	Explanation	Corrective Action
- You Cannot Set Points/Limits For That Device--Try Another MDEX% = 37	The operator has attempted to set analog values for a device which is not analog.	Select an analog device, new function, or CANCEL ACTION.
- You Must Use DISPLAY/DIAGRAM To Get A Diagram Before You Can Use This Command MDEX% = 38	The operator has selected either the START/ENABLE, STOP/DISABLE, SET POINT/LIMITS, AUTO/MANUAL, or MODIFY SCHED function when no data environment diagram is displayed.	Touch the DISPLAY DIAGRAM key and then the desired menu item to display the DE diagram. Now reenter the desired command.
- Higher Level Is Disabled, No Action Taken. MDEX% = 30	The operator has attempted to either start or enable a device, or to obtain a reading from a device which has been logically disconnected from the system.	Enable all devices higher than selected device.
- Data Environment Must Be In Manual Mode Before This Function Can Be Selected. MDEX% = 44	The operator has attempted to either START/ENABLE, STOP/DISABLE, or change the SET POINT/LIMITS on a point on a Data Environment which is in the Automatic Mode of operation.	Put the Data environment into the Manual Mode of Operation.
- Device Is Non-Selectable; Please Try Another MDEX% = 45	The operator has touched a device on the CRT screen which is not either a Function Key, cyan bordered symbol, Menu Box, or Alarm Indicator.	Aim more carefully. HINT: Use the eraser end of a pencil. <u>Do NOT</u> use any metal object on the surface of the screen. (The operator may receive quite a jolt and the system may crash).
- AUTO/MANUAL Function Still Pending MDEX% = 46	The operator has not completed the command sequence. CONFIRM ACTION or CANCEL ACTION must be touched to remove the system from the suspense state.	Touch either CONFIRM ACTION or CANCEL ACTION function key.

Error Message	Explanation	Corrective Action
- Value Out of Range, Please Re-enter MDEX% = 52	During Schedule Modification the operator has entered a value for the schedule entry which is not within an acceptable range, (.i.e., a multiplier value † 9.99; a season other than [S]ummer or [W]inter; a non-standard time; or a temperature outside of design limits).	Re-enter a reasonable value for the schedule entry in the exact format as it is displayed on the screen.
- Unknown Touch Target, Please Try Again MDEX% = 53	The operator has touched an area on the screen other than one of the cyan colored schedule entries.	Aim more carefully. Use the eraser end of a pencil. As a last resort calibrate the Touch Panel.

ALARMS - WHITE ON RED FLASH LIST

Alarm Message	Explanation
- Alarm Acknowledged--Audible Tone Temporarily Silenced MDEX% = 39	This cue appears when the operator touches the alarm indicator to temporarily silence the audible alarm tone. It signifies that the system has received operator acknowledgment of the alarm.
- Alarm Condition Corrected At (x)	This message signals the operator that the alarm condition no longer exists for point x, where x is a point in the system which was previously in alarm.
- Alarm Condition Detected MDEX% = 40	When an alarm condition is passed by EMCS, the MMI alarm indicator will be activated, an audible tone will sound, and this message will appear in the text area.
- To Temporarily Silence Tone And Acknowledge Alarm, Touch The Alarm Indicator MDEX% = 41	This cue appears after the operator has been notified that the alarm condition exists. If the operator so desires, the alarm indicator may to touched to temporarily silence the audible tone.
- Touch DISPLAY DIAGRAM For More Alarm Information $DEX% = 42	This Cue appears after the operator has been notified that an alarm condition exists. If the operator touches the DISPLAY DIAGRAM key it will cause the menu of data environments to be displayed. This menu will show which area has the alarm, and the operator may then command the computer to display the alarm area for subsequent corrective action.

APPENDIX C

Support Software

C-1 Real-Time Clock Driver

C-2 Touch Panel Driver

C-3 Touch Panel Calibration Program

C-4 Scenario Modification Program

C-1

Real-Time Clock Driver

```
;*********************************************************************
;*********************************************************************
;*                                                                  *
;*             REAL TIME CLOCK DEVICE HANDLER ROUTINES              *
;*                                                                  *
;*                  Written by J. Steven Cathcart                   *
;*                                                                  *
;*                  Written  10/16/81       Updated 11/19/81        *
;*                  Revised  12/29/81       Finalized 4/27/82       *
;*                                                                  *
;*  Written for: Chromatics CG Series Color Graphics Computer       *
;*               Model 3999                                         *
;*                                                                  *
;*  Required options:  GPIO Board and Real Time Clock               *
;*                                                                  *
;*  Required Hardware Changes:  Insert Jumper EE on GPIO board      *
;*                                                                  *
;*********************************************************************
;*********************************************************************
;*                                                                  *
;*                                                                  *
;* The following listing consists of two program segments.  The first *
;* segment is the MAIN program used to initialize COUNTER #0 in the  *
;* INTEL 8253 IC on the General Purpose Input/Output (GPIO) board in *
;* the Chromatics CG 3999 Series computer.  The Main program performs *
;* the following tasks:                                             *
;*                                                                  *
;*     (1) Inserts a vector to the starting address of the user     *
;*         supplied interrupt handler (INTHAN) at IR12 vector       *
;*         locations 3B16H (LSB) AND 3B17H (MSB).                   *
;*                                                                  *
;*     (2) Enables the IR12 mask bit by reading memory location     *
;*         IMASK (at address 3B49H) and setting bit 4 to a logic 1, *
;*         outputting the new word to I/O Port MASK (PORT 47H), and *
;*         rewriting the new MASK word back to location IMASK.      *
;*                                                                  *
;*     (3) Configures the Counter/Timer for mode 0 (Interrupt on    *
;*         Terminal Count mode).  In this mode, the 8253 output     *
;*         will initially be low (logic 0) after the mode set oper- *
;*         ation; after the count is loaded into the selected Count *
;*         register (here Counter #0), the output will remain low   *
;*         and the counter will count [down].  When Terminal Count  *
;*         (last count) is reached, the output will go high (logic 1) *
;*         and remain high until the selected Count register (Counter *
;*         #0 here) is reloaded with the Mode or a new count is     *
;*         loaded.  The Counter continues to decrement after the    *
;*         Terminal count has been reached.                         *
;*                                                                  *
;* The second segment is the Interrupt Service Routine (ISR) for    *
;* the Real-Time Clock.  It performs the following functions:       *
;*                                                                  *
;*     (1) Saves the contents of the AF and HL register pairs on    *
;*         the system stack and restores them prior to exiting the  *
;*         Interrupt Service Routine.                               *
;*                                                                  *
;*     (2) Increments the memory location SECNDS (seconds) every    *
;*         pass through the ISR.  It then checks to see if 60       *
```

```
;*          seconds have passed and, if so, increments the memory      *
;*          location MINTES (minutes) and clears the memory location   *
;*          SECNDS (seconds).  Next the ISR checks to see if 60        *
;*          minutes have passed and, if so, increments the memory      *
;*          location HOURS and clears the memory location MINTES       *
;*          (minutes).  Finally the ISR checks to see if 24 hours      *
;*          have passed and, if so, clears the memory location HOURS,  *
;*          and sets a flag (memory location SMPH to 1) so that the    *
;*          BASIC program knows it is time to update the date to the   *
;*          next calendar day.                                         *
;*                                                                     *
;*      (3) Resets the Interrupting Counter output back to the idle    *
;*          -low (logic 0) state prior to re-enabling interrupts.      *
;*          This is done by rewriting to the Counter during counting.  *
;*          Writing the 1st byte stops current counting.  Writing      *
;*          the 2nd byte starts the new count.  The Mode is also       *
;*          rewritten to the counter.                                  *
;*                                                                     *
;*    Important note:  The accuracy of this Real Time Clock will be    *
;*    off a few seconds every day.  The reason for this being that     *
;*    the frequency we are driving the down counter with does not      *
;*    provide us with an integral number of counts per second.  Hence, *
;*    if the Counter/Timer chip were strapped to a higher frequency    *
;*    and a larger value placed into the count register a more accur-  *
;*    ate count can be obtained and therefore greater accuracy on the  *
;*    Real Time Clock.  For purposes of demonstration and this pro-    *
;*    ject the accuracy is adequate and acceptable.                    *
;*                                                                     *
;***********************************************************************
;***********************************************************************
;*
;*   Memory and Input/Output (I/O) Port assignments
;*   for the Initialization (MAIN) program and the
;*   Interrupt Service Routine (ISR) for the Real
;*   Time Clock.
;*
VECTOR      EQU     3B16H     ;ADDRESS OF IR12 INTERUPT VECTOR
IMASK       EQU     3B49H     ;MEMORY LOCATION IMASK
MASK        EQU       47H     ;I/O PORT MASK
WCW         EQU       57H     ;WRITE CONTROL WORD TO 8253 PROGRAMMABLE TIMER
RLCO        EQU       54H     ;READ/LOAD COUNTER #0 I/O PORT ADDRESS
;*
;*   Begin MAIN program here to configure and initialize the 8253
;*   Programmable Timer chip Counter #0 on the GPIO board.
;*
            ORG    04300H     ;PUT PROGRAMS IN UPPER FUNCTION KEY MEMORY
            PUSH     AF       ;SAVE THESE REGISTERS ON THE STACK
            PUSH     HL
            LD     HL,INTHAN  ;GET ADDRESS OF INTERUPT SERVICE ROUTINE
            LD     (VECTOR),HL ;AND SET IR12 POINTER TO START OF ISR
            LD     A,(IMASK)  ;GET CURRENT INTERUPT MASK WORD
            SET      4,A      ;SET BIT 4 OF THE MASK TO LOGIC 1 = ENABLE
            OUT    (MASK),A   ;LATCH THE NEW MASK WORD INTO THE I/O PORT
            LD     (IMASK),A  ;AND REWRITE THE NEW WORD BACK TO MEMORY
            LD       A,30H    ;LOAD THE CONTROL WORD INTO THE ACCUMULATOR
            OUT    (WCW),A    ;OUTPUT CONTROL WORD TO THE 8253 CHIP
            LD       A,12H    ;OUTPUT LSB TO COUNTER #0
```

```
          OUT   (RLCO),A        ;AT I/O PORT 54H
          LD    A,00H           ;OUTPUT MSB OF COUNT TO 8253 TIMER
          OUT   (RLCO),A        ;AT I/O PORT 54H
          POP      HL
          POP      AF
          RET                   ;RETURN TO BASIC
;*
;*  Define RAM - Temporary storage locations for program use
;*
HOURS     DFS        1          ;HOLDS THE HOUR OF THE DAY IN MILITARY FORMAT
MINTES    DFS        1          ;HOLDS THE NUMBER OF MINUTES PAST THE HOUR
SECNDS    DFS        1          ;  "    "    "   "  SECONDS  "    "  MINUTE
SMPH      DFS        1          ;SEMAPHORE MEMORY TO TELL MAIN PROGRAM THAT
;                               ;WE ARE INTO THE NEXT DAY.
;*
;*  Interrupt Service Routine for the Real Time Clock
;*
INTHAN    PUSH     AF           ;PUT A AND F REGISTERS ON THE SIACK
          PUSH     HL           ; " H  "  L     "     "    "    "
          LD       A,30H        ;LOAD CONTROL WORD WITH MODE INTO ACCUMULATOR
;                               ;SET FOR COUNTER #0, LOAD MSB & LSB, INTERUPT
;                               ;ON LAST COUNT, COUNT IN BINARY FORMAT.
          OUT    (WCW),A        ;OUTPUT TO COUNTER #0
          LD       A,12H        ;LOAD COUNT INTO A REGISTER
          OUT    (RLCO),A       ;OUTPUT TO 8253 TIMER
          LD       A,00H        ;LOAD MSB OF COUNT INTO 8253 TIMER
          OUT    (RLCO),A       ;OUTPUT TO CHIP ON GPIO BOARD
          LD     A,(SECNDS)     ;PUT THE NUMBER OF SECONDS INTO THE A REG.
          INC      A            ;ADD 1 TO THE NUMBER OF SECONDS
          LD     (SECNDS),A     ;RESTORE INCREMENTED SECONDS TO MEMORY
          CP       3CH          ;HAVE WE REACHED 60 SECONDS YET?
          JP     M,TIME         ;NO: THEN JUMP TO TIME
          XOR       A           ;YES: THEN CLEAR THE A REG. AND
          LD     (SECNDS),A     ;RESET THE NO. OF SECONDS TO ZERO
          LD     A,(MINTES)     ;NOW LOAD THE NUMBER OF MINUTES INTO A.
          INC      A            ;AND ADD 1 TO IT (SINCE WE REACHED 60 SEC.)
          LD    (MINTES),A      ;RESTORE INCREMENTED MINUTES TO MEMORY
          CP       3CH          ;HAVE WE REACHED 60 MINUTES YET?
          JP     M,TIME         ;NO: THEN JUMP TO TIME
          XOR       A           ;YES: THEN CLEAR THE ACCUMULATOR
          LD     (MINTES),A     ;AND RESET MINUTES TO ZERO
          LD     A,(HOURS)      ;PUT THE NUMBER OF HOURS INTO A NOW.
          INC      A            ;ADD 1 TO IT (SINCE WE REACHED 60 MINUTES).
          LD     (HOURS),A      ;RESTORE INCREMENTED HOURS TO MEMORY
          CP       18H          ;HAVE WE REACHED 24 HOURS YET?
          JP     M,TIME         ;NO: THEN JUMP TO TIME
          XOR       A           ;YES: THEN CLEAR THE A REG. AND
          LD     (HOURS),A      ;RESET THE NUMBER OF HOURS TO ZERO
          LD       A,1          ;PUT A 1 INTO THE ACCUMULATOR
          LD     (SMPH),A       ;AND THEN INTO SMPH MEMORY TO SET FLAG
TIME      POP      HL           ;RESTORE H AND L REGISTERS
          POP      AF           ;   "    A  "  F    "
          EI                    ;RE-ENABLE INTERRUPTS - Z80 CLEARED THEM
          RETI                  ;RETURN CONTROL TO INTERRUPTED ROUTINE
```

C-2

Touch Panel Driver

```
;***************************************************************************
;***************************************************************************
;*                                                                         *
;*                   TOUCH PANEL DEVICE HANDLER ROUTINES                   *
;*                                                                         *
;*                   Written by:  J. Steven Cathcart                       *
;*                                                                         *
;*                           Written 4/23/82                               *
;*                                                                         *
;*  Written for: Chromatics CG Series Color Graphics Computers Model 3999  *
;*                                                                         *
;*  Required Option: Serial Input/Output Port #1                           *
;*                                                                         *
;***************************************************************************
;***************************************************************************
;*                                                                         *
;*  This program consists of two routines: a MAIN routine and an INTERRUPI *
;*  SERVICE ROUTINE (ISR).                                                 *
;*                                                                         *
;*                                                                         *
;*  IMPORTANT NOTE:  The documentation from Chromatics, the manufacturer   *
;*  of the CG 3999 Color Graphics Computer, on Using Interrupts from       *
;*  Assembly Language is incorrect.  As of 4/26/82 their documentation     *
;*  showed SIO #1 having a higher priority than SIO #0. This is incorrect. *
;*  The correct priorities are:                                            *
;*                                                                         *
;*       MASK BIT    VECTOR       HIGHEST PRIORITY            IRxx         *
;*                                                                         *
;*          7         3B10H       FLOPPY SERVICE              IR15         *
;*          6         3B12H       FLOPPY DATA                 IR14         *
;*  >>>>    5         3B14H       SERIAL PORT #0 DATA RCVD    IR13    <<<< *
;*          4         3B16H       COUNTER/TIMER CLOCK CHIP    IR12         *
;*  >>>>    3         3B18H       SERIAL PORT #1 DATA RCVD    IR11    <<<< *
;*          2         3B1AH       LIGHT PEN HIT               IR10         *
;*          1         3B1CH       ERASE PAGE COMPLETE         IR9          *
;*          0         3B1EH       KEYBOARD STROBE             IR8          *
;*                                                                         *
;*                                LOWEST PRIORITY                          *
;*                                                                         *
;***************************************************************************
;***************************************************************************
;*                                                                         *
;*  The MAIN routine performs the following tasks:                         *
;*                                                                         *
;*  (1)  Inserts a vector to the starting address of the user supplied     *
;*       interrupt handler (LATCH) at IR11 Vector locations 3B18H (LSB)    *
;*       and 3B19H (MSB).                                                  *
;*                                                                         *
;*  (2)  Enables the IR11 Mask bit by reading memory location 3B49H        *
;*       (IMASK) and setting bit 3 to a logic 1 - enable. Then outputting  *
;*       the new mask word to I/O Port 47H (MASK), and rewriting the new   *
;*       interrupt mask word back to location 3B49H (IMASK).               *
;*                                                                         *
;*  (3)  Initializes all memory locations used by interrupt service        *
;*       routine to zero.  The FLAG is also initially set to zero.         *
;*                                                                         *
;* ENTRY:  Called from BASIC as an Assembly Language Subroutine            *
```

```
;*                                                                          *
;* EXIT:   Returns program control back to MMIINIT program segment          *
;*                                                                          *
;* CALLS:  None                                                             *
;*                                                                          *
;****************************************************************************
;*                                                                          *
;* Set up values to be used in program by assembler using EQU statements   *
;*                                                                          *
;****************************************************************************
;****************************************************************************
       ORG 04280H
VECTOR EQU  3B18H    ;ADDRESS OF IR12 INTERRUPT VECTOR
IMASK  EQU  3B49H    ;MEMORY LOCATION IMASK
MASK   EQU    47H    ;I/O PORT MASK
CLSNG  EQU    7BH    ;CLOSING CODE FROM LOGIC BOX - 123
UNCVR  EQU    7CH    ;UNCOVER CODE FROM LOGIC BOX - 124
SIO1D  EQU    4EH    ;SIO 1 DATA PORT
SIO1S  EQU    4FH    ;SIO 1 STATUS PORT
;*
;* MAIN PROGRAM - REWRITE IR11 INTERRUPT VECTOR INTO MEMORY LOCATIONS
;* VECTOR (3B18H-3B19H) AND ENABLE INTERRUPT FROM SERIAL I/O PORT #1
;* BY SETTING MASK BIT 3 TO A 1 IN MEMORY LOCATION IMASK (3B49H) AND
;* WRITING THIS NEW MASK BYTE OUT TO I/O PORT MASK (47H).
;*
MAIN    LD   HL,LATCH        ;GET ADDRESS OF INTERRUPT SERVICE ROUTINE
        LD   (VECTOR),HL     ;AND SET IR11 POINTER TO START OF ISR
        LD   A,(IMASK)       ;GET CURRENT INTERRUPT MASK WORD
        SET   3,A            ;SET BIT 5 OF MASK TO LOGIC 1 = ENABLE
        OUT  (MASK),A        ;LATCH THE NEW MASK WORD INTO THE I/O PORT
        LD   (IMASK),A       ;AND REWRITE THE NEW WORD BACK TO MEMORY
        LD   A,00H           ;INITIALIZE TEMPORARY MEMORY AND SEMAPHORE
        LD   (FLAG),A        ;FLAG MEMORY LOCATIONS
        LD   (TEMP),A
        LD   (XTUCH),A       ;ALSO INITIALIZE COORDINATE MEMORY LOCATIONS
        LD   (YTUCH),A
        EI
        RET                  ;RETURN TO BASIC
;*
;* DEFINE RAM VARIABLE STORAGE
;*
FLAG    DFS  1              ;SEMAPHORE FLAG FOR BASIC
TEMP    DFS  1              ;TEMPORARY STORAGE LOCATION
XTUCH   DFS  1              ;X TOUCH COORDINATE FROM LOGIC BOX
YTUCH   DFS  1              ;Y TOUCH COORDINATE FROM LOGIC BOX
;****************************************************************************
;****************************************************************************
;*                                                                          *
;*                     LATCH INTERRUPT SERVICE ROUTINE                      *
;*                                                                          *
;*  For each touch panel hit the Carroll Touch Panel Logic Box generates   *
;*  four codes.  These four codes are passed to the Chromatics CG 3999     *
;*  Computer over a serial input/output line.  They enter the CG Computer  *
;*  through Serial Input/Output Port #1 (SIO #1).  The initial baud rate   *
;*  of SIO #1 is set to 150 baud.  This interrupt service routine should   *
;*  work with any baud rate up to 9600 baud.  As each code is received by  *
;*  the 8251 USART attached to SIO #1 the USART causes an interrupt to the *
```

```
;*   processor to occur.  Hence, each Touch Panel hit will cause four inter-*
;*   rupts in sequence.  In order to buffer a Touch Panel hit these four    *
;*   interrupts must occur in the following order -- Uncover code - X coor- *
;*   dinate - Y coordinate - Closing code --.  As these four interrupts     *
;*   occur in order the ISR will know what to do with each one of them be-  *
;*   cause of the way it is written.                                        *
;*                                                                          *
;*   ENTRY:  Interrupt from 8251 USART on SIO #1                            *
;*                                                                          *
;*   EXIT:   Returns program control back to BASIC                          *
;*                                                                          *
;*   CALLS:  None                                                           *
;*                                                                          *
;****************************************************************************
;****************************************************************************
;*                                                                          *
;*   The interrupt service routine performs the following tasks:            *
;*                                                                          *
;*   (1)   Saves the contents of the AF and HL register pairs on the Stack   *
;*         and restores them prior to exiting the ISR.                      *
;*                                                                          *
;*   (2)   Inputs the data directly from the 8251 USART - I/O Port SIO1D     *
;*         (Port 4EH) - and strips off bit 7 to force the input to be a     *
;*         positive number in the range from 0 to 127 decimal.              *
;*                                                                          *
;*   (3)   Checks memory lcation FLAG to determine if the Executive routine  *
;*         from BASIC has picked up the previous Touch Panel input.  If it  *
;*         has then the latest input is accepted, else input is ignored and *
;*         the ISR is exitted.                                              *
;*                                                                          *
;*   (4)   Once input is accepted the first value from the Carroll logic    *
;*         box is checked.  It should be a 124 decimal (UNCOVER code).      *
;*         When the Uncover code is detected the ISR then puts a value of   *
;*         120 decimal into the XTUCH and YTUCH memory locations.           *
;*                                                                          *
;*         NOTE:  Any value above 110 decimal is out of the Carroll Touch   *
;*                Panel°s range - i.e. 0 <= X coordinate <= 110  and        *
;*                0 <= Y coordinate <= 94.  Hence, we use values outside    *
;*                of its range but less than 128 decimal as identifiers -   *
;*                i.e. 124 decimal identifies the beginning of the file or  *
;*                information; 120 decimal identifies that a value has not  *
;*                been input yet for this coordinate; and 123 decimal iden- *
;*                tifies the end of file or information.                    *
;*                                                                          *
;*   (5)   The second and third inputs from the logic box should then be the *
;*         X and Y Touch Panel coordinates, respectively.  The X coordinate *
;*         is placed into memory first and then the Y coordinate is buffered.*
;*                                                                          *
;*   (6)   Finally, the fourth interrupt should input the Closing code to   *
;*         the ISR.  Once the Closing code is detected the ISR sets the     *
;*         memory location FLAG to a 1 (ONE) - i.e. raises the FLAG for     *
;*         BASIC to see that a Touch Panel hit has been received and buffer- *
;*         ed into memory.  The ISR is then exitted.                        *
;*                                                                          *
;****************************************************************************
;****************************************************************************
;*
```

```
;*
;* LATCH SUBROUTINE - THIS USER SUPPLIED SUBROUTINE WILL BUFFER THE
;* TOUCH PANEL COORDINATES SENT FROM THE LOGIC BOX OF THE CARROLL
;* TOUCH PANEL .  IT FIRST CHECKS TO SEE THAT THE EXECUTIVE ROUTINE
;* OF THE BASIC PROGRAM HAS SEEN THE PREVIOUS TOUCH PANEL HIT BEFORE
;* IT ACCEPTS A NEW COORDINATE PAIR FROM THE LOGIC BOX.
;*
LATCH   PUSH AF             ;SAVE THESE REGISTERS ON THE SIACK
        PUSH HL             ;AND RESTORE AT END OF ROUTINE
        IN   A,(SIO1D)      ;NOW INPUT THE DATA FROM THE SERIAL PORT
        AND  7FH            ;FORCE INPUT INTO 7 BITS
        LD   (TEMP),A       ;SAVE CURRENT INPUT FROM SIO1D IN MEMORY
        LD   A,(FLAG)       ;NOW CHECK TO SEE IF THE BASIC PROGRAM HAS
;                           ;SEEN THE LAST TOUCH PANEL HIT AND RESET FLAG
        CP   00H            ;IS THE FLAG = 0?
        JR   NZ,EXIT-$      ;NO-THEN IGNORE THIS INPUT
        LD   A,(TEMP)       ;YES-THEN CHECK CURRENT INPUT
        CP   UNCVR          ;IS CURRENT INPUT THE UNCOVER CODE?
        JR   Z,L01-$        ;YES-THEN JUMP TO PREPARE FOR COORDINATES
        CP   CLSNG          ;NO-THEN IS CURRENT INPUT THE CLOSING CODE?
        JR   Z,L02-$        ;YES-THEN JUMP TO SET FLAG
        LD   A,(XTUCH)      ;NO-THEN BY DEFAULT INPUT IS A COORDINATE
        CP   120            ;IS THIS THE X COORDINATE?
        JP   M,YIN          ;NO-THEN JUMP TO CHECK FOR Y COORDINATE
        LD   A,(TEMP)       ;YES-PUT X COORDINATE INTO MEMORY
        LD   (XTUCH),A
        JR   EXIT-$         ;THEN EXIT SUBROUTINE
YIN     LD   A,(YTUCH)      ;INPUT SHOULD BE A Y COORDINATE
        CP   120            ;BUT CHECK ANYWAY
        JP   M,EXIT         ;NO-IF NOT Y COORDINATE THEN EXIT SUBROUTINE
        LD   A,(TEMP)       ;YES-PUT Y COORDINATE INTO MEMORY
        LD   (YTUCH),A
        JR   EXIT-$         ;EXIT SUBROUTINE
L01     LD   A,120          ;WE HAVE THE UNCOVER CODE - RESET X AND Y
        LD   (XTUCH),A      ;TO THE VALUE 120 (i.e. PUT UNIQUE VALUE
        LD   (YTUCH),A      ;INTO THESE LOCATIONS)
        JR   EXIT-$         ;EXIT ROUTINE
L02     LD   A,1            ;SET FLAG = 1
        LD   (FLAG),A
EXIT    POP  HL             ;RESTORE THESE REGISTERS BEFORE RETURNING
        POP  AF
        EI                  ;RE-ENABLE INTERRUPTS SINCE Z80 CLEARED THEM
        RETI                ;RETURN TO WHERE WE INTERRUPTED FROM
```

C-3

Touch Panel Calibration Program

Touch Panel Calibration Program

```
10 REM TOUCH PANEL CALIBRATOR
20 REM 1/25/82 - RCC & JSC
25 DIM TX(121), TY(121), LX(121), LY(121), CAL(8)
30 PRINT CHR$(27);"R13";CHR$(27);"S1";CHR$(98);
40 PRINT ""M"2"C0"N"C1";CHR$(12);
45 PRINT ""C7"U132495 TOUCH PANEL CALIBRATION - REV 1.00 - RCC"
50 PRINT ""W050041471470"C2"R";CHR$(12);
55 PRINT : PRINT
60 PRINT "This program is to be used to calibrate the MMI Touch
     Panel Input"
70 PRINT "routines so that touch panel target coordinates can be
     correctly"
80 PRINT "mapped onto absolute screen coordinates."
90 PRINT : PRINT
100 PRINT "This program should be run whenever the display unit
     has been moved"
110 PRINT "or the touch panel reasembled.  If the program is not
     run following"
120 PRINT "such a move or reassemblage, then the MMI software
     may not correctly"
130 PRINT "read the touch panel input."
140 PRINT : PRINT
150 PRINT "You will be asked to touch each of 121 blue-filled
     circles (targets)"
160 PRINT "in succession.  As you touch each circle it will
     change color and an"
169 PRINT "audible tone will sound.  Work carefully, as the
     accuracy of the "
170 PRINT "recalibration depends upon how accurately you hit the
     targets.  A "
180 PRINT "pencil (eraser tip toward screen) will prove helpful
     in hitting the "
190 PRINT "targets." : PRINT
200 PRINT : LINEINPUT ""M"1"C1"N"2"C7 - Depres The RETURN Key
     When Ready To Begin - ";Q$
220 PRINT ""=";CHR$(12);""K"G*"F";
230 FOR X=4 TO 512 STEP 50
240 FOR Y=4 TO 512 STEP 50
250 PRINT ""C1"; : PLOT X,Y,3 : C=C+1
275 TI%=1
300 ON ERROR #4 GOTO 600
310 GOTO 310
315 ON ERROR #0 GOTO 0
320 LX(C)=X : TX(C)=T%(2) : LY(C)=Y : TY(C)=T%(3)
330 PRINT ""C6"; : PLOT X,Y,3 : PRINT CHR$(7);
340 NEXT Y
350 NEXT X
370 PRINT CHR$(21);CHR$(12);""C5CALCULATING CALIBRATION FACTORS
     . . . PLEASE WAIT . . ."
380 DEFDBL X,Y,B,D,A,Z
385 N=121
```

Touch Panel Calibration Program

```
390 X=0:Y=0:B0=0:B1=0:B2=0:B3=0:J=0:X1=0:X2=0:X3=0:X4=0:X5=0:Y1=0
    :Y2=0:Y3=0
395 'DECLARING A USER DEFINED FUNCTION FOR SOLVING THE
    DETERMINATE OF A 3x3 MATRIX
400 DEF FNDET(A1,A2,A3,A4,A5,A6,A7,A8,A9)=A1*(A5*A9-A6*A8)-A4*(A2
    *A9-A3*A8)+A7*(A2*A6              -A3*A5)
410 FOR I = 1 TO N
420 J=J+1:X=X+TX(I):Y=Y+LX(I):X1=X1+TX(I)^2:X2=X2+TX(I)^3:X3=X3+T
    X(I)^4
430 X4=X4+TX(I)^5:X5=X5+TX(I)^6:Y1=Y1+LX(I)*TX(I)
440 Y2=Y2+LX(I)*TX(I)^2:Y3=Y3+LX(I)*TX(I)^3
450 NEXT I
455 GOSUB 700
456 CAL(1)=B0 : CAL(2)=B1 : CAL(3)=B2 : CAL(4)=B3
460 X=0:Y=0:B0=0:B1=0:B2=0:B3=0:J=0:X1=0:X2=0:X3=0:X4=0:X5=0:Y1=0
    :Y2=0:Y3=0
465 FOR I = 1 TO N
470 J=J+1:X=X+TY(I):Y=Y+LY(I):X1=X1+TY(I)^2:X2=X2+TY(I)^3:X3=X3+T
    Y(I)^4
475 X4=X4+TY(I)^5:X5=X5+TY(I)^6:Y1=Y1+LY(I)*TY(I)
480 Y2=Y2+LY(I)*TY(I)^2:Y3=Y3+LY(I)*TY(I)^3
485 NEXT I
490 GOSUB 700
492 CAL(5)=B0 : CAL(6)=B1 : CAL(7)=B2 : CAL(8)=B3
495 PRINT : PRINT : PRINT "~C3NEW FACTORS ARE:" : PRINT
496 PRINT "X:",CAL(1),CAL(2),CAL(3),CAL(4) : PRINT
497 PRINT "Y:",CAL(5),CAL(6),CAL(7),CAL(8) : PRINT
500 PRINT : PRINT "~C6SAVING NEW CALIBRATION FACTORS TO DISK . .
    ."
510 DOS "ARYSAVE TPCALDAT CAL"
520 PRINT : PRINT : PRINT "~C2CALIBRATION COMPLETE"
530 END
600 IF ERR=24 THEN T%(TI%)=INP(&H4E) ELSE ON ERROR#0 GOTO 0
610 IF T%(TI%)=123 THEN RESUME 315 : TI%=1 ELSE TI%=TI%+1
620 RESUME
640 'SUBROUTINE USED TO SET UP AND SOLVE 4 EQUATIONS WITH 4
    UNKNOWNS USING
650 'CRAMER'S RULE SOLUTION TO SOLVE FOR THE COEFFICIENTS OF THE
    THIRD ORDER
660 'MAPPING FUNCTION.  IT IS NECESSARY TO USE A THIRD ORDER
    MAPPING FUNCTION
670 'BECAUSE OF THE NON-LINEARITY OF THE TOUCH PANEL.
700 Z1 = FNDET(X1,X2,X3,X2,X3,X4,X3,X4,X5)
710 Z2 = FNDET(X,X1,X2,X2,X3,X4,X3,X4,X5)
720 Z3 = FNDET(X,X1,X2,X1,X2,X3,X3,X4,X5)
730 Z4 = FNDET(X,X1,X2,X1,X2,X3,X2,X3,X4)
740 DD = J*Z1-X*Z2+X1*Z3-X2*Z4
750 Z1 = FNDET(X1,X2,X3,X2,X3,X4,X3,X4,X5)
760 Z2 = FNDET(X,X1,X2,X2,X3,X4,X3,X4,X5)
770 Z3 = FNDET(X,X1,X2,X1,X2,X3,X3,X4,X5)
780 Z4 = FNDET(X,X1,X2,X1,X2,X3,X2,X3,X4)
790 D0 = Y*Z1-Y1*Z2+Y2*Z3-Y3*Z4
```

Touch Panel Calibration Program

```
800 Z1 = FNDET(Y1,X2,X3,Y2,X3,X4,Y3,X4,X5)
810 Z2 = FNDET(Y,X1,X2,Y2,X3,X4,Y3,X4,X5)
820 Z3 = FNDET(Y,X1,X2,Y1,X2,X3,Y3,X4,X5)
830 Z4 = FNDET(Y,X1,X2,Y1,X2,X3,Y2,X3,X4)
840 D1 = J*Z1-X*Z2+X1*Z3-X2*Z4
850 Z1 = FNDET(X1,Y1,X3,X2,Y2,X4,X3,Y3,X5)
860 Z2 = FNDET(X,Y,X2,X2,Y2,X4,X3,Y3,X5)
870 Z3 = FNDET(X,Y,X2,X1,Y1,X3,X3,Y3,X5)
880 Z4 = FNDET(X,Y,X2,X1,Y1,X3,X2,Y2,X4)
890 D2 = J*Z1-X*Z2+X1*Z3-X2*Z4
900 Z1 = FNDET(X1,X2,Y1,X2,X3,Y2,X3,X4,Y3)
910 Z2 = FNDET(X,X1,Y,X2,X3,Y2,X3,X4,Y3)
920 Z3 = FNDET(X,X1,Y,X1,X2,Y1,X3,X4,Y3)
930 Z4 = FNDET(X,X1,Y,X1,X2,Y1,X2,X3,Y2)
940 D3 = J*Z1-X*Z2+X1*Z3-X2*Z4
950 B0 = D0/DD: B1 = D1/DD: B2 = D2/DD: B3 = D3/DD
960 RETURN
```

C-4

Scenario Modification Program

Scenario Modification Program

Energy Monitoring and Control System
Man/Machine Interface (EMCS/MMI)

```
10 REM SCENARIO MODIFICATION PROGRAM
20 NRTPRTS% = 93 : NNSIM% = 28
30 DIM SIM(NRTPTS%)
31 PRINT CHR$(12);""™=";
32 PRINT ""™M™C1™1™N™C7MMI SCENARIO MODIFICATION - REV
     1.01™M™2™C0™N™C2"
33 PRINT : PRINT
34 PRINT ""™W000000511481™R";
40 INPUT ""™C2Please Enter The Number of the Time Step to be
     Modified => ™C6"; K%
50 PRINT
60 IF K% > NNSIM% OR K% < 1 THEN PRINT CHR$(7);""™C4There is Not a
     Time Step Number™C3";K%;""™C4. The Largest Time Step Number is
     ™C328™C4.™C2" : PRINT : GOTO 40
70 DOS "ARYLOAD SIM"+MID$(STR$(K%),2)+" SIM"
80 INPUT ""™C2Please Enter the Number of the EMCS Point to be
     Modified in this Time Step => ™C6"; J%
90 PRINT
100 IF J% > NRTPTS% OR J% < 1 THEN PRINT CHR$(7);""™C4There is Not
     a EMCS Point Number™C3";J%;""™C4. The Largest EMCS Point
     Number is ™C393™C4.™C2" : PRINT : GOTO 80
110 PRINT ""™C2Please Type in the New Value For Point Number™C3";
120 PRINT " (";J%;") ™C2=>™C6 "; : INPUT A
130 SIM(J%) = A
140 PRINT
150 LINEINPUT ""™C2Do You Wish To Modify Another Point in this Time
     Step (Y/N) ? ™C6"; B$
160 PRINT
170 IF LEN(B$)>0 AND (LEFT$(B$,1)="Y" OR LEFT$(B$,1)="N") THEN IF
     LEFT$(B$,1)="Y" THEN B$="" : J%=0 : GOTO 80 ELSE B$="" : J%=0
     ELSE PRINT CHR$(7);""™C4Please Answer Yes or No.™C2" : PRINT :
     GOTO 150
180 DOS "ARYSAVE SIM"+MID$(STR$(K%),2)+" SIM"
190 LINEINPUT ""™C2Do You Wish To Modify Another Time Step (Y/N) ?
     ™C6"; C$
200 PRINT
210 IF LEN(C$)>0 AND (LEFT$(C$,1)="Y" OR LEFT$(C$,1)="N") THEN IF
     LEFT$(C$,1)="Y" THEN C$="" : K%=0 : GOTO 40 ELSE C$="" : K%=0
     ELSE PRINT CHR$(7);""™C4Please Answer Yes or No.™C2" : PRINT :
     GOTO 190
220 LINEINPUT ""™C2Do You Want a Complete Listing of the Scenario
     (Y/N) ? ™C6"; D$
230 PRINT : IF LEN(D$)>0 AND (LEFT$(D$,1)="Y" OR LEFT$(D$,1)="N")
     THEN IF LEFT$(D$,1)="Y" THEN D$="" : GOTO 250 ELSE D$="" ELSE
     PRINT CHR$(7);""™C4Please Answer Yes or No.™C2" : PRINT : GOTO
     220
240 PRINT : PRINT ""™M™C1™1™N™C7End of Scenario Modification.™="; :
```

Scenario Modification Program

Energy Monitoring and Control System
Man/Machine Interface (EMCS/MMI)

```
    END
250 REM PRINT SIMULATION ARRAYS
260 LAST%=28 : NUMDEV%=93  'LAST% IS MAX SIM #  -  NUMDEV% IS MAX
    DEVICE #
270 DIM MS(LAST%,NUMDEV%)
280 PRINT CHR$(27);"R07";                  ' SET PRINTER BAUD RATE TO
                                             1200 BAUD
290 PRINT
300 LINEINPUT ""C2Please Turn On The Printer and Align Paper. . .
    Press RETURN To Continue . . .";A$
310 PRINT
320 PRINT "Loading Data Base . . . Please Wait . . ."
330 FOR X%=1TOLAST%
340 DOS "ARYLOAD SIM"+MID$(STR$(X%),2)+" SIM"
350 FOR Y%=1 TO NUMDEV% : MS(X%,Y%)=SIM(Y%) : NEXT Y%
360 NEXT X%
370 PRINT : PRINT "Printing Data . . . Please Wait . . ." : PRINT
380 NDIVS%=LAST%\7 : IF LAST%MOD7<>0 THEN NDIVS%=NDIVS%+1
                                           ' CALCULATE MAX # DIVS
390 PG%=0 : GOTO 490
400 LPRINT : LPRINT
410 LPRINT TAB(16);"Table Of Real-Time Device Simulation Values --
    PAGE";PG%
420 LPRINT : LPRINT
430 LPRINT TAB(39);"Simulation Time Period"
440 LPRINT
450 LPRINT TAB(10);"DEVICE#  ";:FOR X%=START% TO FINISH%:LPRINT
    USING "     ####";X%;:NEXT X%:LPRINT:LPRINT:LPRINT
460 FOR Y%=TP% TO BOT%:LPRINT:LPRINT:LPRINT USING "
    ###";Y%;:LPRINT "            ";:FOR X%=START%TOFINISH%:LPRINT
    USING "###.#   ";MS(X%,Y%);:NEXT X%,Y%
470 LPRINT CHR$(12);
480 RETURN
490 FOR TP%=1 TO NUMDEV% STEP 24 : BOT%=TP%+23 : IF BOT%>NUMDEV%
    THEN BOT%=NUMDEV%
500 FINISH%=0
510 FOR DIV%=1 TO NDIVS%
520 START%=FINISH%+1 : FINISH%=START%+6 : IF FINISH%>LAST% THEN
    FINISH%=LAST%
530 PG%=PG%+1 : GOSUB 400
540 NEXT DIV%
550 NEXT TP%
560 LPRINT: LPRINT CHR$(12);
580 END
```

APPENDIX D

Beginning Line Numbers for Modules

APPENDIX D

Beginning Line Numbers for Modules

MODULE	LINE NUMBER
ADISABLE	9000
ANNUNC	2200
AUTO	3900
BUZOFF	3300
CANCEL	4600
CONFIRM	4300
DIAGRAM	3600
DISMESS	7100
DMENU	7200
DRAWIT	9200
DSCHED	11100
FKEYIN	2400
GJTS	110
GONOGO	3400
HELP	1500
IJTS	160
INITDB	400
KBISR	70
MDIN	4700
MENUIN	3100
MLIN	4800
MMIEXEC	10
MMIEXEC (subroutine Entry Point)	200
MOIN	5100
MRIN	5200
MSCHED	11300
MSIN	5300
OPER	4200
REPORT	4000
REPOUT	5400
SCHED	4100

Beginning Line Numbers for Modules (continued)

MODULE	LINE NUMBER
SETPT	3700
SYMBIN	2600
TIME	7400
TOUCHIN	2100
UPDATE	1800
XAM	6400
XCHOP	6800
XDD	6100
XMS	6700
XPR	6500
XSD	5800
XSE	5500
XSPL	6300
ZZZZ (Last Line Number)	11800

APPENDIX E

EMCS/MMI Documented Code

Initialization Program Segment
Annotated Source Code Listing

Energy Monitoring and Control System
Man/Machine Interface (EMCS/MMI)
Initial Release - May 1982

```
1 REM MMIINIT - System Initialization Program Segment
2 REM ======= (semi-condensed form -- annotated)
3 REM
4 REM GEORGIA INSTITUTE OF TECHNOLOGY
5 REM ENGINEERING EXPERIMENT STATION, COMMAND & CONTROL DIVISION
6 REM Development Team:  B.B.Wise, R.C.Coleburn, J.S.Cathcart, B.S.Rice
7 REM Initial Release - REV 1.00 - May 1982
8 REM
10 GOTO 280
50 STOP
60 REM KB ISR (Keyboard ISR) ================================================
70 IF ERR=24 THEN T%=INP(&H4A) ELSE ON ERROR #0 GOTO 0
80 IF T%=8 THEN IF LEN(ZIN$)>0 THEN PRINT CHR$(8);" ";CHR$(8); : ZIN$=LEFT(ZIN$,LEN(ZIN$)-1) : RESUME
90 IF T%<>13 THEN IF T%<32 OR T%>126 THEN RESUME ELSE PRINT CHR$(T%); : ZIN$=ZIN$+CHR$(T%) : RESUME ELSE KISR%=1 :
      PRINT : RESUME
199 REM STRIPED DOWN EXEC ================================================
200 IQ%=1 : ON ERROR #1 GOTO 70
260 IF KISR%=1 THEN IQ%=0 : KISR%=0 : ON ERROR #0 GOTO 0 : RETURN
270 GOTO 260
280 GOSUB 400                                                    ' Call InitDB
290 DOS "DRAW MMIPRE" : PRINT ""="K";CHR$(12);CHR$(27);"W"?050";       ' Draw initial screen.
295 GOSUB 7400                                                   ' Call TIME
300 PRINT : PRINT ""C2Starting Up Touch Panel I/O Processor . . ."
310 DOS "FETCH TPDRIVER.ABS 4280"                                '  Load in assembly level program.
320 DEF USR1=&H4280 : C=USR1(0)                                  ' Define entry point and start up TPDRIVER.
330 DOS "DRAW MMILOGO" : PRINT CHR$(12);CHR$(28);""="K";CHR$(27);"W"?050";
                                                                 ' Draw logo.
335 GOSUB 1500                                                   ' Call HELP
340 MC%=1 : DOS "CHAIN MMIMAIN"                                  ' Return to main program segment.
350 STOP
400 REM INITDB ===========================================================
402 CLEAR 2050 : PRINT ""M"C0"2"N"C2"K";CHR$(12);"MMI/EMCS DEMO DEVICE - REV 1.00 - APRIL, 1982" : PRINT : PRINT :
      PRINT "Loading DataBase . . . Please Wait . . ."
407 T%=0 : KISR%=0 : ZIN$=""
408 CJT#=0 : IJT#=0 : Y%=0 : D%=0 : H%=0 : M%=0 : S%=0 : MT%=0 : TPT%=0 : M1%=0 : M2%=0 : UC#=0 : UT%=100 : TC#=0
409 DIM JDAY%(12,1) : DOS "ARYLOAD JDAY JDAY%" : Y9%=0 : D9%=0 : H9%=0 : M9%=0 : S9%=0 : MQUE%=5 : TQUE%=0 : BQUE%=0 :
      DIM AQUE#(1,MQUE%)
410 DOS "OPEN 5 R MESSAGE 100"
420 NHELPBF%=12 : NMESS%=54 : NRTPTS%=93 : NKEYS%=10 : NOPER%=5 : NUMREPTS%=3 : NDES%=3 : NSE%=25
430 DIM FKLOC%(NKEYS%), PX%(NRTPTS%), PY%(NRTPTS%)
440 DIM PTYPE%(NRTPTS%), PHIER%(NRTPTS%)
```

Initialization Program Segment
Annotated Source Code Listing

Energy Monitoring and Control System
Man/Machine Interface (EMCS/MMI)
Initial Release - May 1982

```
.
450 DIM SSED%(NRTPTS%), CDBVL(NRTPTS%), NWVL(NRTPTS%)
460 DIM NBOXES%(7) : DOS "ARYLOAD NBOXES NBOXES%" : DOS "ARYLOAD SSED SSED%"
470 DOS "ARYLOAD FKLOC FKLOC%" : DOS "ARYLOAD PX PX%" : DOS "ARYLOAD PY PY%"
480 DOS "ARYLOAD PTYPE PTYPE%" : DOS "ARYLOAD PHIER PHIER%"
520 PAUSEALRM%=0 : SALARM%=0
530 HC%=0 : SHUTDOWN%=0 : RODS%=0 : ZC%=0
540 GDTYPE%=-1 : LDE%=0 : DIAMEN%=0 : FDEAM%=1
560 DBRDY%=0 : APT%=0 : DPT%=0 : ANPT%=0
570 QFCN%=0 : QPT%=0 : QMENU%=0 : QAM%=-1 : QSCH%=0 : QOPER%=0
580 DIMCAL(8) : DOS "ARYLOAD TPCALDAT CAL"
605 PRINT CHR$(27);"R13";CHR$(27);"S1";CHR$(98);
610 DIM SID$(23), KTEXT$(3,NKEYS%), KCOLR%(2,NKEYS%), BOXLOC%(3), TT$(7,3)
620 SID$(0)="FID" : SID$(1)="MUX1" : SID$(2)="MUX2" : SID$(3)="R" : SID$(4)="S" : SID$(5)="A" : SID$(6)="E" :
      SID$(7)="F" : SID$(8)="FL" : SID$(9)="LV"
630 SID$(10)="M" : SID$(11)="O" : SID$(12)="PS" : SID$(13)="T" : SID$(14)="W" : SID$(15)="DP" : SID$(16)="STR" :
      SID$(17)="TC" : SID$(18)="HW BOILER" : SID$(20)="CHILLER"
650 KTEXT$(1,1)="START/" : KTEXT$(2,1)="" : KTEXT$(3,1)="ENABLE"
660 KTEXT$(1,2)="STOP/" : KTEXT$(2,2)="" : KTEXT$(3,2)="DISABLE"
670 KTEXT$(1,3)="DISPLAY" : KTEXT$(2,3)="" : KTEXT$(3,3)="DIAGRAM"
680 KTEXT$(1,4)="SET" : KTEXT$(2,4)="POINT/" : KTEXT$(3,4)="LIMITS"
690 KTEXT$(1,5)="AUTO/" : KTEXT$(2,5)="" : KTEXT$(3,5)="MANUAL"
700 KTEXT$(1,6)="PRINT" : KTEXT$(2,6)="" : KTEXT$(3,6)="REPORT"
710 KTEXT$(1,7)="MODIFY" : KTEXT$(2,7)="" : KTEXT$(3,7)="SCHED"
720 KTEXT$(1,8)="CHANGE" : KTEXT$(2,8)="" : KTEXT$(3,8)="OPER"
730 KTEXT$(1,9)="CONFIRM" : KTEXT$(2,9)="" : KTEXT$(3,9)="ACTION"
740 KTEXT$(1,10)="CANCEL" : KTEXT$(2,10)="" : KTEXT$(3,10)="ACTION"
750 BOXLOC%(1)=340 : BOXLOC%(2)=260 : BOXLOC%(3)=180
760 TT$(1,0)="MENU OF DATA ENVIRONMENTS" : TT$(1,1)="BUILDING 106"
770 TT$(1,2)="MAIN CHILLER PLANT" : TT$(1,3)="SUPPLY NO. 2"
780 TT$(2,0)="MENU OF ANALOG CONTROL SET POINT/LIMITS"
790 TT$(2,1)="LOW OPERATING LIMIT" : TT$(2,2)="SET POINT"
800 TT$(2,3)="HIGH OPERATING LIMIT"
810 TT$(3,0)="MENU OF ANALOG MONITOR SET POINT/ LIMITS" : TT$(3,1)="LOW ALARM LIMIT"
820 TT$(3,3)="HIGH ALARM LIMIT" : TT$(3,2)="SET POINT"
830 TT$(4,0)="MENU OF REPORTS"
840 TT$(4,1)="ENERGY UTILIZATION SUMMARY"
850 TT$(4,2)="LOCK OUT SUMMARY" : TT$(4,3)="ALARM SUMMARY"
860 TT$(5,0)="MENU OF OPERATIONS" : TT$(5,1)="CHANGE OPERATOR"
870 TT$(5,2)="HELP" : TT$(5,3)="SYSTEM SHUTDOWN"
880 TT$(6,0)="MENU OF OUTPUT DEVICES"
890 TT$(6,1)="LINEPRINTER" : TT$(6,2)="CRT SCREEN"
910 TT$(7,0)="MENU OF SCHEDULES"
```

Initialization Program Segment
Annotated Source Code Listing

Energy Monitoring and Control System
Man/Machine Interface (EMCS/MMI)
Initial Release - May 1982

```
920 TT$(7,1)="BUILDING 106 - Optimum Start/Stop Schedule"
930 TT$(7,2)="MAIN CHILLER PLANT - Optimum Start/Stop Schedule"
940 TT$(7,3)="SUPPLY NO. 2 - Optimum Start/Stop Schedule"
970 NCNT%=5 : NALARM%=10 : NIND%=9 : LCNT%=0 : LALARM%=5 : LIND%=15
980 DOS "ARYLOAD KCOLR KCOLR%"
990 DIM OPER$(NOPER%), PWRD$(NOPER%), OLEVEL%(NOPER%)
1000 OPER$(1)="COLEBURN" : OPER$(2)="CATHCART" : OPER$(3)="WISE"
1010 OPER$(4)="CANFIELD" : OPER$(5)="RICE"
1020 PWRD$(1)="RANDY" : PWRD$(2)="STEVE" : PWRD$(3)="BILLY" : PWRD$(4)="KARLIN" : PWRD$(5)="BEV"
1030 OLEVEL%(1)=3 : OLEVEL%(2)=3 : OLEVEL%(3)=3 : OLEVEL%(4)=3 : OLEVEL%(5)=3
1040 NNSIM%=28 : SIM%=0
1100 DIM ADPT%(NRTPTS%), LOWLIM(NRTPTS%), HILIM(NRTPTS%), SPT(NRTPTS%), ALARMVL%(NRTPTS%)
1110 DIM ASTAT%(NRTPTS%), DSTAT%(NRTPTS%), LPNT%(NRTPTS%)
1120 DOS "ARYLOAD ADPT ADPT%" : DOS "ARYLOAD LOWLIM LOWLIM" : DOS "ARYLOAD HILIM HILIM"
1130 DOS "ARYLOAD SPT SPT" : DOS "ARYLOAD LPNT LPNT%"
1135 FOR I%=1 TO NRTPTS% : ALARMVL%(I%)=0 : NEXT
1136 ALARMVL%(27)=1 : ALARMVL%(59)=1 : ALARMVL%(64)=1
1140 FOR I%=1 TO NRTPTS% : DSTAT%(I%)=0 : ASTAT%(I%)=0 : CDBVL(I%)=0 : NEXT I%
1150 DIM SCHTMP$(NSE%), S$(NDES%,NSE%), VPS%(1,NSE%)
1151 DOS "ARYLOAD VPS VPS%"
1152 S$(1,1)="68.2F" : S$(1,2)="0730" : S$(1,3)="78.1F" : S$(1,4)="89.4F" : S$(1,5)="CLG"
1153 S$(1,6)="NA" : S$(1,7)="0800" : S$(1,8)="0800" : S$(1,9)="0800" : S$(1,10)="0800"
1154 S$(1,11)="0800" : S$(1,12)="NA" : S$(1,13)="NA" : S$(1,14)="NA" : S$(1,15)="1700"
1155 S$(1,16)="1700" : S$(1,17)="1700" : S$(1,18)="1700" : S$(1,19)="1700" : S$(1,20)="NA"
1156 S$(1,21)="NA" : S$(1,22)="S" : S$(1,23)="65F" : S$(1,24)="80F" : S$(1,25)="5.00"
1160 FOR I%=2 TO NDES% : FOR X%=1 TO NSE% : S$(I%,X%)=S$(1,X%) : NEXT X%,I%
1170 DIM PNAME$(23,2)
1180 PNAME$(0,0)="DISABLED" : PNAME$(0,1)="ENABLED" : PNAME$(0,2)="FID"
1190 PNAME$(1,0)="DISABLED" : PNAME$(1,1)="ENABLED" : PNAME$(1,2)="MUX #1"
1200 PNAME$(2,0)="DISABLED" : PNAME$(2,1)="ENABLED" : PNAME$(2,2)="MUX #2"
1210 PNAME$(3,0)="DEGREES F." : PNAME$(3,1)="" : PNAME$(3,2)="Temperature Controller Reset Interface"
1220 PNAME$(4,0)="STOPPED" : PNAME$(4,1)="STARTED" : PNAME$(4,2)="Start/Stop Control Interface"
1230 PNAME$(5,0)="NO ALARM" : PNAME$(5,1)="ALARM DETECTED" : PNAME$(5,2)="Alarm Contact Signal Monitor"
1240 PNAME$(6,0)="???" : PNAME$(6,1)="" : PNAME$(6,2)="Enthalpy/Economizer Control Interface"
1250 PNAME$(7,0)="GPM" : PNAME$(7,1)="" : PNAME$(7,2)="Flow Indication Monitor"
1260 PNAME$(8,0)="NO FLAME" : PNAME$(8,1)="FLAME PRESENT" : PNAME$(8,2)="Flame Indication Monitor"
1270 PNAME$(9,0)="LOW" : PNAME$(9,1)="HIGH" : PNAME$(9,2)="Level Indication Monitor"
1280 PNAME$(10,0)="KW" : PNAME$(10,1)="" : PNAME$(10,2)="Meter"
1290 PNAME$(11,0)="OFF/NONE/OPEN" : PNAME$(11,1)="ON/POSITIVE/CLOSED" : PNAME$(11,2)="On/Off Status Monitor"
1300 PNAME$(12,0)="PERCENT OPEN" : PNAME$(12,1)="" : PNAME$(12,2)="Position Sensor"
1310 PNAME$(13,0)="Degrees F." : PNAME$(13,1)="" : PNAME$(13,2)="Temperature Monitor"
1320 PNAME$(14,0)="%RH" : PNAME$(14,1)="" : PNAME$(14,2)="Humidity Monitor"
```

Initialization Program Segment
Annotated Source Code Listing

Energy Monitoring and Control System
Man/Machine Interface (EMCS/MMI)
Initial Release - May 1982

```
1330 PNAME$(15,0)="OPEN" : PNAME$(15,1)="CLOSED" : PNAME$(15,2)="Differential Pressure Switch"
1340 PNAME$(16,0)="STOPPED" : PNAME$(16,1)="STARTED" : PNAME$(16,2)="Motor Starter"
1350 PNAME$(17,0)="DEGREES F." : PNAME$(17,1)="" : PNAME$(17,2)="Temperature Controller"
1360 PNAME$(18,0)="STOPPED" : PNAME$(18,1)="STARTED" : PNAME$(18,2)="Hot Water Boiler"
1370 PNAME$(19,0)="STOPPED" : PNAME$(19,1)="STARTED" : PNAME$(19,2)="Pump"
1380 PNAME$(20,0)="STOPPED" : PNAME$(20,1)="STARTED" : PNAME$(20,2)="Chiller"
1390 PNAME$(21,0)="STOPPED" : PNAME$(21,1)="STARTED" : PNAME$(21,2)="Cooling Tower"
1400 PNAME$(22,0)="STOPPED" : PNAME$(22,1)="STARTED" : PNAME$(22,2)="Unit Air Conditioner"
1410 PNAME$(23,0)="NOT IN USE" : PNAME$(23,1)="IN USE" : PNAME$(23,2)="Coil"
1420 RETURN
1500 REM HELP ===============================================================
1510 HC%=1                                                      ' Set flag that help is running.
1530 H$="HLP"+MID$(STR$(HC%),2) : DOS "DRAW "+H$                ' Retrieve help buffer file.
1550 PRINT CHR$(12);CHR$(28);""="K";CHR$(27);"W"J";             ' Display on screen.
1560 MDEX%=24 : GOSUB 7100 : ZIN$="" : GOSUB 200 : HC%=HC%+1    ' Wait for operator response, then advance to
                                                                  next screen.

1620 IF ZIN$<>"S" AND HC%<=NHELPBF% GOTO 1530                   ' Continue until last screen displayed.
1650 HC%=0 : DOS "DRAW MMISCRN" : PRINT CHR$(12);CHR$(28);""=";CHR$(27);"W";
                                                                ' Turn off flag, redraw operator screen.
1660 PRINT ""W396502510492"M"2"C6"N"2"C0"P"K";CHR$(27);"0A1";   ' Define time window.
1670 PRINT ""W001130510451"M"2"C0"N"2"C1"P"K";CHR$(27);"0A2";   ' Define graphics window.
1680 PRINT ""W001051510128"M"2"C0"N"2"C1"P"K";CHR$(27);"0A3";   ' Define function key window.
1690 PRINT ""W001001510049"M"2"C0"N"2"C2"R"J";                  ' Define text window.
1700 GDTYPE%=-1 : RETURN                                        ' Done.
7100 REM DISMESS ============================================================
7110 DOS "ACCESS 5"+STR$(MDEX%) : LINEINPUT #5;MESS$ : PRINT MESS$;""M"C0"2"N"C2" : RETURN
                                                                ' Retrieve and display message.
7400 REM TIME ===============================================================
7410 DOS "FETCH CLOCK.ABS 4300"                                 ' Load in assembly level subroutine.
7420 DEF USR0=&H4300                                            ' Define entry point.
7430 PRINT CHR$(12);""M"C1"1"N"C7"J - INITIALIZATION - "M"2"C0"N"
7440 PRINT : PRINT : PRINT ""C5ENTER THE CURRENT DATE AS FOLLOWS:"
7450 PRINT : INPUT ""C3ENTER THE MONTH (2 DIGITS) ";MT%
7455 IF MT%>12 THEN GOSUB 8000 : GOTO 7450
7460 PRINT : INPUT "ENTER THE DAY   (2 DIGITS) ";D%
7465 IF D%>31 THEN GOSUB 8000 : GOTO 7460
7470 PRINT : INPUT "ENTER THE YEAR  (2 DIGITS) ";Y%
7475 IF Y%>99 THEN GOSUB 8000 : GOTO 7470
7490 PRINT : PRINT : PRINT : PRINT ""C5ENTER THE CURRENT TIME AS FOLLOWS:"
7500 PRINT : INPUT ""C3ENTER THE HOUR (MILITARY, 2 DIGITS) ";H%
7505 IF H%>23 THEN GOSUB 8000 : GOTO 7500
```

Initialization Program Segment
Annotated Source Code Listing

Energy Monitoring and Control System
Man/Machine Interface (EMCS/MMI)
Initial Release - May 1982

```
7510 PRINT : M$="" : INPUT "ENTER THE MINUTE (2 DIGITS) ";M%
7515 IF M%>59 THEN GOSUB 8000 : GOTO 7510
7520 PRINT : INPUT "ENTER THE SECOND (2 DIGITS) ";S%
7525 IF S%>59 THEN GOSUB 8000 : GOTO 7520
7530 POKE &H4321,H%                                          ' Store current hour.
7540 POKE &H4322,M%                                          ' Store current minute.
7550 POKE &H4323,S%                                          ' Store current second.
7560 PRINT : PRINT ""C2Starting Up Clock . . ." : C=USRO(0)  ' Start-up clock.
7590 M2%=D% : IF Y%MOD4=0 THEN M1%=1 ELSE M1%=0              ' Determine if leap year.
7600 D%=D%+JDAY%(MT%-1,M1%)                                  ' Get current # days elapsed this year.
7610 CJ#=Y%*1000000000#+D%*1000000#+H%*10000#+M%*100#+S%     ' Compute Julian Date.
7611 TC#=CJ# : UC#=CJ#                                       ' Set initial time control values.
7620 RETURN
8000 PRINT ""C4"11-ERROR - Incorrect Time/Date Argument, Please Re-Enter!"C3"2"
8010 RETURN
```

Main Program Segment
Annotated Source Code Listing

Energy Monitoring and Control System
Man/Machine Interface (EMCS/MMI)
Initial Release - May 1982

```
1 REM MMIMAIN - Main Program Segment
2 REM ======= (semi-condensed form -- annotated)
3 REM
4 REM GEORGIA INSTITUTE OF TECHNOLOGY
5 REM ENGINEERING EXPERIMENT STATION, COMMAND & CONTROL DIVISION
6 REM Development Team:  B.B.Wise, R.C.Coleburn, J.S.Cathcart, B.S.Rice
7 REM Initial Release - REV 1.00 - May 1982
8 REM
10 IF MC%=0 THEN DOS "CHAIN MMIINIT"                                   ' Initial start-up.
30 IF MC%=2 GOTO 250 ELSE GOTO 190                                     ' Returning from MMIPART2.
69 REM KB ISR (Keyboard ISR) ============================================
70 IF ERR=24 THEN T%=INP(&H4A) ELSE ON ERROR #0 GOTO 0
80 IF T%=8 THEN IF LEN(ZIN$)>0 THEN PRINT CHR$(27);"0A3";CHR$(8);" ";CHR$(8);CHR$(27);"0A0"; :
      ZIN$=LEFT$(ZIN$,LEN(ZIN$)-1) : RESUME
90 IF T%<>13 THEN IF T%<32 OR T%>126 THEN RESUME ELSE PRINT CHR$(27);"0A3";CHR$(T%);CHR$(27);"0A0"; :
      ZIN$=ZIN$+CHR$(T%) : RESUME ELSE KISR%=1 : PRINT CHR$(27);"0A3"; : PRINT  : PRINT CHR$(27);"0A0"; : RESUME
100 MDEX%=28 : GOSUB 7100 : STOP                                       ' Fail-Safe.
109 REM GJTS ============================================================
110 H%=PEEK(&H4321) : M%=PEEK(&H4322) : S%=PEEK(&H4323) : IF PEEK(&H4324)=1 THEN GOTO 120 ELSE
      CJT#=Y%*1000000000#+D%*1000000#+H%*10000#+M%*100#+S% : RETURN
120 D%=D%+1 : POKE&H4324,0 : IF Y%MOD4=0 THEN TP%=366 : M1%=1 ELSE TP%=365 : M1%=0
130 Y%=Y%+D%\TP% : D%=D%MODTP% : MT%=0 : IF Y%>99 THEN MDEX%=28 : GOSUB 7100
140 MT%=MT%+1 : IF D%>JDAY%(MT%,M1%) THEN 140
150 CJT#=Y%*1000000000#+D%*1000000#+H%*10000#+M%*100#+S% : IF MT%>1 THEN M2%=D%MODJDAY%(MT%-1,M1%) : RETURN ELSE
      M2%=D% : RETURN
159 REM IJTS ============================================================
160 S9%=S%+IN% : M9%=M%+S9%\60 : S9%=S9%MOD60 : H9%=H%+M9%\60 : M9%=M9%MOD60 : D9%=D%+H9%\24 : H9%=H9%MOD24 : IF
      Y%MOD4=0 THEN TP%=366 ELSE TP%=365
170 Y9%=Y%+D9%\TP% : D9%=D9%MODTP% : IF Y9%>99 THEN MDEX%=28 : GOSUB 7100
180 IJT#=Y9%*1000000000#+D9%*1000000#+H9%*10000#+M9%*100#+S9% : RETURN
189 REM EXEC ============================================================
190 GOSUB 1800 : MDEX%=11 : GOSUB 7100 : ON ERROR #0 GOTO 0 : GOTO 201
                                                                       ' Initial entry point.
200 TISR%=0 : IQ%=1 : ON ERROR #1 GOTO 70                              ' Keyboard enable entry point.
201 IF HC%>0 GOTO 220 ELSE GOSUB 110                                   ' Do not display time when help running.
210 PRINT CHR$(27);"0A0";CHR$(28); : PRINT USING"**/";MT%,M2%; : PRINT USING"**  ";Y%; : PRINT USING"** : ";H%,M%; :
      PRINT USING"**";S%; : PRINT CHR$(27);"0A3";                      ' Update time display.
220 IF BQUE%=TQUE% GOTO 240 ELSE IF AQUE#(1,TQUE%)>CJT# GOTO 240       ' Check alarm queue.
230 DSTAT%(AQUE#(0,TQUE%))=0 : TQUE%=TQUE%+1 : IF TQUE%=MQUE% THEN TQUE%=0
240 IF FDEAM%=1 AND CJT#>UC# AND IQ%<>1 THEN GOSUB 1800                ' Check if ready to update.
250 MC%=1 : IF PEEK(&H42A0)=1 AND HC%=0 AND IQ%=0 THEN GOSUB 2100 : POKE&H42A0,0
```

Main Program Segment
Annotated Source Code Listing

Energy Monitoring and Control System
Man/Machine Interface
Initial Release - May 1982

```
                                                                          ' Check for touch panel hit.
260 IF KISR%=1 THEN IQ%=0 : KISR%=0 : ON ERROR #0 GOTO 0 : PRINT CHR$(27);"OA3"; : RETURN
                                                                          ' Check for keyboard input.
270 GOSUB 110 : IF SALARM%=1 AND PAUSEALRM#<=CJT# AND TC#<=CJT# THEN INC%=5 : GOSUB 160 : TC#=IJT# : PRINT CHR$(7);
                                                                          ' Check if ready to sound alarm.
280 IF SHUTDOWN%<>1 GOTO 201 ELSE ON ERROR #0 GOTO 0 : POKE&H4220,1      ' Check for shutdown.
290 DOS "DRAW SHUTDOWN" : PRINT CHR$(27);"OA1"M"CO"N";CHR$(12);CHR$(27);"W";CHR$(27);"OA3";
300 MDEX%=16 : GOSUB 7100 : NEW : END
1799 REM UPDATE ============================================================
1800 SIM%=SIM%+1 : IF SIM%>NNSIM% THEN SIM%=1                            ' Read in next simulation array.
1820 DOS "ARYLOAD SIM"+MID$(STR$(SI%),2)+" NW" : TM%=1
1840 IF DSTAT%(TM%)=1 GOTO 1930                                           ' Ignore disabled points.
1850 IF NWVL(TM%)=CDBVL(TM%) GOTO 1930                                    ' Dont need to process points whose values have
                                                                            not changed.
1860 CDBVL(TM%)=NWVL(TM%) : IF SSED%(TM%)=1 AND GDTYPE%=1 AND DIAMEN%=LPNT%(TM%) THEN TP%=QPT% : QPT%=TM% : D1%=1 :
     D2%=3 : D3%=0 : GOSUB 9150 : QPT%=TP%                                ' Re-color point on screen if necessary.
1870 IF ABS(PTYPE%(TM%))<LALARM% OR ABS(PTYPE%(TM%))>=LALARM%+NALARM% GOTO 1930
                                                                          ' Can point generate an alarm.
1880 IF ADPT%(TM%)<>0 GOTO 1970                                           ' Analog or digital.
1890 IF (CDBVL(TM%)<LOLIM(TM%)) OR (CDBVL(TM%)>HILIM(TM%)) GOTO 1980     ' Check for analog alarm.
1900 IF ASTAT%(TM%)<>1 GOTO 1930                                          ' Was point previously in alarm.
1910 ASTAT%(TM%)=0 : ANPT%=TM% : GOSUB 2200                               ' Annunciate change.
1930 TM%=TM%+1 : IF TM%<=NRTPTS% GOTO 1840                                ' Loop for all points.
1950 DBRDY%=1 : GOSUB 110 : INC%=UT% : GOSUB 160 : UC#=IJT# : RETURN     ' Update time flag.
1970 IF CDBVL(TM%)<>ALARMVL%(TM%) GOTO 1900                               ' Check for digital alarm.
1980 IF ASTAT%(TM%)<>0 GOTO 1930                                          ' Was point previously not in alarm.
1990 ASTAT%(TM%)=1 : APT%=TM% : ANPT%=TM% : GOSUB 2200 : GOTO 1930       ' Annunciate change.
2099 REM TOUCHIN ===========================================================
2100 CX%=PEEK(&H42A2) : CY%=PEEK(&H42A3)                                  ' Get coordinates from TPDRIVER.
2111 XTUCH%=FIX(CAL(1)+CAL(2)*CX%+CAL(3)*CX%^2+CAL(4)*CX%^3)             ' Compute screen coordinates
2112 YTUCH%=FIX(CAL(5)+CAL(6)*CY%+CAL(7)*CY%^2+CAL(8)*CY%^3)
2120 IF XTUCH%<0 OR XTUCH%>511 THEN IF XTUCH%>511 THEN XTUCH%=511 ELSE XTUCH%=0
                                                                          ' Validate X
2130 IF YTUCH%<0 OR YTUCH%>511 THEN IF YTUCH%>511 THEN YTUCH%=511 ELSE YTUCH%=0
                                                                          ' Validate Y
2140 IF YTUCH%<=116 AND YTUCH%>=63 THEN GOSUB 2400 : RETURN              ' Check for function key hit.
2150 IF YTUCH%>452 AND XTUCH%<=122 THEN GOSUB 3300 : RETURN              ' Check for alarm indicator hit.
2160 IF YTUCH%>452 OR YTUCH%<129 THEN RETURN                              ' Ignore invalid hits.
2170 IF GDTYPE%=1 THEN GOSUB 2600 : RETURN                                ' Check for symbol hit.
2180 IF GDTYPE%=0 THEN GOSUB 3100 : RETURN                                ' Check for menu hit.
2190 MDEX%=28 : GOSUB 7100 : RETURN                                       ' OOPS.
2199 REM ANNUNC ============================================================
```

Main Program Segment
Annotated Source Code Listing

Energy Monitoring and Control System
Man/Machine Interface
Initial Release - May 1982

```
2200 D3%=0 : IF SSED%(ANPT%)=1 THEN IF CD(ANPT%)=1 THEN D3%=D3%+2 ELSE D3%=D3%+0 ELSE IF DSTAT%(ANPT%)=0 THEN D3%=D3%+2
                                                                                ' Determine point color.
2210 IF ASTAT%(ANPT%)=1 GOTO 2260                                               ' Is it a new alarm.
2220 PRINT ""M"C4"1"N"C7-Alarm Condition Corrected At:  ";PNAME$(ABS(PTYPE%(ANPT%)),2);""M"C0"2"N"
2230 IF GDTYPE%=1 AND DIAMEN%=LPNT%(ANPT%) THEN D1%=1 : D2%=3 : TP%=QPT% : QPT%=ANPT% : GOSUB 9200 : QPT%=ANPT%
                                                                                ' Redraw point if necessary.
2240 IF APT%=ANPT% THEN D1%=3 : GOSUB 9200 : SALARM%=0 : APT%=0                 ' Turn off alarm indicator if necessary.
2250 RETURN
2260 D1%=2 : GOSUB 9200 : SALARM%=1 : PAUSEALRM#=0 : TC#=0                      ' Turn on alarm indicators.
2280 PRINT ""M"C4"1"N"C7-ALarm Condition Detected At:  ";PNAME$(ABS(PTYPE%(ANPT%)),2);""M"C0"2"N"
2290 IF GDTYPE%=1 AND DIAMEN%=LPNT%(ANPT%) THEN D1%=1 : D2%=3 : D3%=D3%+8 : TP%=QPT% : QPT%=ANPT% : GOSUB 9200 :
     QPT%=ANPT% ELSE MDEX%=42 : GOSUB 7100                                      ' Redraw point if necessary.
2300 RETURN
2399 REM FKEYIN ============================================================
2400 IF QFCN%>0 THEN D1%=0 : D2%=1 : D3%=0 : GOSUB 9200 : IF XTUCH%<FKLOC%(9) THEN MDEX%=7 : GOSUB 7100 : QFCN%=0 :
     ZC%=0                                                                      ' Deselect any previous function.
2420 IF XTUCH%<FKLOC%(2) THEN QFCN%=1 : GOSUB 3400 : RETURN                     ' START/ENABLE.
2430 IF XTUCH%<FKLOC%(3) THEN QFCN%=2 : GOSUB 3400 : RETURN                     ' STOP/DISABLE.
2440 IF XTUCH%<FKLOC%(4) THEN GOSUB 3600 : RETURN                               ' DISPLAY DIAGRAM.
2450 IF XTUCH%<FKLOC%(5) THEN GOSUB 3700 : RETURN                               ' SET/POINT LIMITS.
2460 IF XTUCH%<FKLOC%(6) THEN GOSUB 3900 : RETURN                               ' AUTO/MANUAL.
2490 IF XTUCH%<FKLOC%(9) THEN DOS "CHAIN MMIPART2"                              ' All others in MMIPART2.
2500 IF XTUCH%<FKLOC%(10) THEN GOSUB 4300 : RETURN                              ' CONFIRM.
2510 GOSUB 4600 : RETURN                                                        ' CANCEL.
2599 REM SYMBIN ============================================================
2600 IF QPT%>0 THEN D1%=1 : D2%=2 : D3%=0 : GOSUB 9160 : MDEX%=6 : GOSUB 7100
                                                                                ' Deselect any previous point.
2620 I%=1                                                                       ' Loop until found which point was touched.
2630 IF DIAMEN%=LPNT%(I%) GOTO 2670
2640 I%=I%+1 : IF I%<=NRTPTS% GOTO 2630
2660 MDEX%=45 : GOSUB 7100 : QPT%=0 : RETURN                                    ' Did not find it.
2670 IF (PX%(I%)<=XTUCH%) AND (XTUCH%<=PX%(I%)+25) THEN 2680 ELSE 2640
2680 IF (PY%(I%)<=YTUCH%) AND (YTUCH%<=PY%(I%)+25) THEN 2690 ELSE 2640
2690 IF ABS(PTYPE%(I%))>=LIND% GOTO 2660                                        ' Check if non-selectable point.
2700 QPT%=I% : D1%=1 : D2%=2 : D3%=1 : GOSUB 9160                               ' Found point - highlight it.
2710 IF ASTAT%(QPT%)=1 THEN TM$=""C4" ELSE TM$=""C2"                            ' Display point status.
2720 IF DSTAT%(QPT%)=1 THEN PRINT ""C2"M"C0"2"N"2";PNAME$(ABS(PTYPE%(QPT%)),2);" Is Disabled" : GOTO 2760
2730 IF ADPT%(QPT%)=1 THEN PRINT ""M"C0"2"N"2"C2";PNAME$(ABS(PTYPE%(QPT%)),2);" - Value:
     ";TM$;PNAME$(ABS(PTYPE%(QPT%)),CD(QPT%));""C2, AlarmVal:  ";PNAME$(ABS(PTYPE%(QPT%)),ALARMVL%(QPT%)) : GOTO 2760
2740 IF ABS(PTYPE%(QPT%))>=LCNT%+NCNT% THEN PRINT ""M"C0"2"N"C2";PNAME$(ABS(PTYPE%(QPT%)),2);" - Value:
     ";TM$;CD(QPT%);""C2 ";PNAME$(ABS(PTYPE%(QPT%)),0);", LoLim: ";LOLIM(QPT%);", HiLim: ";HILIM(QPT%) : GOTO 2760
2750 PRINT ""M"C0"2"N"C2"2";PNAME$(ABS(PTYPE%(QPT%)),2);" - SetPt: ";TM$;SPT(QPT%);"
```

Main Program Segment
Annotated Source Code Listing

Energy Monitoring and Control System
Man/Machine Interface
Initial Release - May 1982

```
      "C2";PNAME$(ABS(PTYPE%(QPT%)),0);", LoLim: ";LOLIM(QPT%);", HiLim: ";HILIM(QPT%)
2760 IF QFCN%>0 GOTO 2820                                   ' Has function been selected.
2770 IF FDEAM%=0 GOTO 2810                                  ' Is DE in manual mode.
2780 D1%=1 : D2%=2 : D3%=0 : GOSUB 9160 : QPT%=0 : RETURN   ' Nope - Deselect point
2810 MDEX%=20 : GOSUB 7100 : RETURN                         ' In manual - Ready to touch function key.
2820 IF QFCN%<>1 GOTO 2890                                  ' Is function START/ENABLE.
2830 IF DSTAT%(QPT%)=1 GOTO 2860                            ' Is point disabled.
2840 IF SSED%(QPT%)=1 AND CDBVL(QPT%)=0 GOTO 2860 ELSE MDEX%=26 : GOSUB 7100 : GOTO 3050
                                                            ' Check if already started.
2860 IF PHIER%(QPT%)>0 THEN 2870 ELSE 3040                  ' Is it a low-level point.
2870 IF DSTAT%(PHIER%(QPT%))=0 THEN IF SSED%(PHIER%(QPT%))=0 OR CDBVL(PHIER%(QPT%))=1 GOTO 3040
                                                            ' Is higher level disabled.
2880 MDEX%=30 : GOSUB 7100 : GOTO 3050                      ' Yes - display error.
2890 IF QFCN%<>2 GOTO 2930                                  ' Is function STOP/DISABLE.
2900 IF DSTAT%(QPT%)=1 GOTO 2920                            ' Is point already disabled.
2910 IF SSED%(QPT%)=1 AND CDBVL(QPT%)=0 GOTO 2920 ELSE GOTO 3040   ' Check if already stopped.
2920 MDEX%=27 : GOSUB 7100 : GOTO 3050                      ' Already disabled - give error.
2930 IF QFCN%<>4 GOTO 2980                                  ' Is function SET POINT/LIMITS.
2940 IF ADPT%(QPT%)=1 GOTO 2970                             ' Is point analog or digital.
2950 IF LALARM%<=ABS(PTYPE%(QPT%)) AND ABS(PTYPE%(QPT%))<LALARM%+NALARM% THEN DIAMEN%=3 ELSE DIAMEN%=2
                                                            ' Is point control or alarm.
2960 GOSUB 7200 : MDEX%=15 : GOSUB 7100 : MDEX%=23 : GOSUB 7100 : RETURN
                                                            ' Display menu.
2970 MDEX%=25 : GOSUB 7100 : GOTO 3050                      ' Can not set digital alarm.
2980 IF QFCN%<>5 GOTO 3030                                  ' Is function AUTO/MANUAL.
2990 MDEX%=46 : GOSUB 7100 : D1%=1 : D2%=2 : D3%=0 : GOSUB 9160 : QPT%=0 : GOTO 3040
                                                            ' AUTO/MANUAL function still pending - Deselect
                                                              point.
3030 MDEX%=28 : GOSUB 7100 : GOTO 3050                      ' OOPS.
3040 MDEX%=18 : GOSUB 7100 : RETURN                         ' Got a function and a point - ready to execute.
3050 D1%=1 : D2%=2 : D3%=0 : GOSUB 9160 : QPT%=0 : MDEX%=17 : GOSUB 7100 : RETURN
                                                            ' Deselect point - Ask for another.
3099 REM MENUIN =========================================================
3100 IF ZC%=1 THEN GOSUB 11300 : RETURN                     ' If MODIFY SCHED running then goto MSCHED.
3120 IF QMENU%=0 GOTO 3160 ELSE D1%=-1 : D2%=1 : D3%=0      ' Deselect any previous menu selection.
3125 IF DIAMEN%=1 AND LPNT%(APT%)=QMENU% THEN D3%=8
3140 GOSUB 9200 : QMENU%=0
3150 MDEX%=8 : GOSUB 7100
3160 IF DIAMEN%=1 THEN GOSUB 4700 : RETURN                  ' DISPLAY DIAGRAM menu.
3170 IF DIAMEN%=2 OR DIAMEN%=3 THEN GOSUB 4800 : RETURN     ' SETPT/LIMITS menu.
3180 REM --- all other menus in MMIPART2 ---
3220 MDEX%=28 : GOSUB 7100 : RETURN                         ' OOPS.
```

Main Program Segment
Annotated Source Code Listing

Energy Monitoring and Control System
Man/Machine Interface
Initial Release - May 1982

```
3299 REM BUZOFF ============================================================
3300 GOSUB 110 : INC%=120 : GOSUB 160 : PAUSEALRM#=IJT# : TC#=IJT# : MDEX%=39 : GOSUB 7100 : RETURN
                                                                     ' Temporarily silence alarm bell.
3399 REM GONOGO ============================================================
3400 D1%=0 : D2%=1 : D3%=1 : GOSUB 9200                              ' Backlight key.
3420 IF GDTYPE%<>1 THEN MDEX%=38 : GOSUB 7100 : D1%=0 : D2%=1 : D3%=0 : GOSUB 9200 : QFCN%=0 : MDEX%=11 : GOSUB 7100 :
     RETURN                                                          ' Make sure DE displayed.
3430 IF FDEAM%=1 THEN MDEX%=44 : GOSUB 7100 : D1%=0 : D2%=1 : D3%=0 : GOSUB 9200 : QFCN%=0 : MDEX%=11 : GOSUB 7100 :
     RETURN                                                          ' Make sure in manual mode.
3440 IF QPT%=0 THEN MDEX%=17 : GOSUB 7100 : RETURN                   ' Has point been selected.
3450 IF QFCN%<>1 GOTO 3540                                           ' Is function START/ENABLE.
3460 IF DSTAT%(QPT%)=1 GOTO 3490                                     ' Is point disabled.
3470 IF SSED%(QPT%)=1 AND CDBVL(QPT%)=0 GOTO 3490                    ' Is point stopped.
3480 MDEX%=26 : GOSUB 7100 : GOTO 3580                               ' Nope - give error.
3490 IF PHIER%(QPT%)<=0 GOTO 3530                                    ' Is it a low-level point.
3500 IF DSTAT%(PHIER%(QPT%))=1 GOTO 3520                             ' Yes - is higher level disabled.
3510 IF SSED%(PHIER%(QPT%))=0 OR CDBVL(PHIER%(QPT%))<>0 GOTO 3530    ' Is higher level not stopped.
3520 MDEX%=30 : GOSUB 7100 : GOTO 3580                               ' Higher level disabled - give error.
3530 MDEX%=18 : GOSUB 7100 : RETURN                                  ' Ready to do it.
3540 IF DSTAT%(QPT%)=1 GOTO 3560                                     ' Is point disabled.
3550 IF SSED%(QPT%)=0 OR CDBVL(QPT%)=1 GOTO 3570                     ' Is point started.
3560 MDEX%=27 : GOSUB 7100 : GOTO 3580                               ' Nope - give error.
3570 MDEX%=18 : GOSUB 7100 : RETURN                                  ' OK - Ready to do it.
3580 D1%=1 : D2%=2 : D3%=0 : GOSUB 9160 : QPT%=0 : MDEX%=17 : GOSUB 7100 : RETURN
                                                                     ' Deselect point.
3599 REM DIAGRAM ===========================================================
3600 QFCN%=3 : D1%=0 : D2%=1 : D3%=1 : GOSUB 9200 : QPT%=0 : QMENU%=0 : DIAMEN%=1 : GOSUB 7200 : MDEX%=21 : GOSUB 7100
     : RETURN                                                        ' Backlight key & draw menu.
3699 REM SETPT =============================================================
3700 QFCN%=4 : D1%=0 : D2%=1 : D3%=1 : GOSUB 9200                    ' Backlight key.
3720 IF GDTYPE%<>1 THEN MDEX%=38 : GOSUB 7100 : D1%=0 : D2%=1 : D3%=0 : GOSUB 9200 : QFCN%=0 : MDEX%=11 : GOSUB 7100 :
     RETURN                                                          ' Make sure DE displayed.
3730 IF FDEAM%=1 THEN MDEX%=44 : GOSUB 7100 : D1%=0 : D2%=1 : D3%=0 : GOSUB 9200 : QFCN%=0 : MDEX%=11 : GOSUB 7100 :
     RETURN                                                          ' Make sure in manual mode.
3750 QMENU%=0 : IF QPT%=0 THEN MDEX%=17 : GOSUB 7100 : RETURN        ' Need to select a device.
3760 IF ADPT%(QPT%)=1 THEN MDEX%=25 : GOSUB 7100 : D1%=1 : D2%=2 : D3%=0 : GOSUB 9160 : QPT%=0 : MDEX%=17 : GOSUB 7100
     : RETURN                                                        ' Determine if control or monitor.
3770 IF LALARM%<=ABS(PTYPE%(QPT%)) AND ABS(PTYPE%(QPT%))<LALARM%+NALARM% THEN DIAMEN%=3 ELSE DIAMEN%=2
                                                                     ' Can not set digital alarm.
3780 GOSUB 7200 : MDEX%=15 : GOSUB 7100 : MDEX%=23 : GOSUB 7100 : RETURN
                                                                     ' OK - Ready to do it.
3899 REM AUTO ==============================================================
```

Main Program Segment
Annotated Source Code Listing

Energy Monitoring and Control System
Man/Machine Interface
Initial Release - May 1982

```
3900 QFCN%=5 : D1%=0 : D2%=1 : D3%=1 : GOSUB 9200                     ' Backlight key.
3920 IF GDTYPE%<>1 THEN MDEX%=38 : GOSUB 7100 : D1%=0 : D2%=1 : D3%=0 : GOSUB 9200 : QFCN%=0 : MDEX%=11 : GOSUB 7100 :
    RETURN                                                              ' Make sure DE displayed.
3930 IF FDEAM%=1 THEN QAM%=0 : PRINT ""C2"2"M"C0"2"N-Manual Mode Selected For DE:  ";TT$(1,DIAMEN%) : GOTO 3950
                                                                        ' Select manual mode.
3940 QAM%=1 : PRINT ""C2"2"M"C0"2"N-Auto Mode Selected For DE:  ";TT$(1,DIAMEN%)
                                                                        ' Select auto mode.
3950 MDEX%=18 : GOSUB 7100 : RETURN                                     ' OK - Ready to do it.
4299 REM CONFIRM ==========================================================
4300 TM%=QFCN% : QFCN%=9 : D1%=0 : D2%=1 : D3%=1 : GOSUB 9200 : QFCN%=TM%
                                                                        ' Backlight Key.
4320 IF (QFCN%<=0) OR (QFCN%>8) THEN 4440                               ' Make sure function selected.
4330 IF QFCN%=1 THEN IF QPT%>=1 THEN GOSUB 5500 : GOTO 4470 ELSE GOTO 4440
                                                                        ' XSE
4340 IF QFCN%=2 THEN IF QPT%>=1 THEN GOSUB 5800 : GOTO 4470 ELSE GOTO 4440
                                                                        ' XSD
4350 IF QFCN%=3 THEN IF QMENU%>=1 AND QMENU%<=NDES% THEN GOSUB 6100 : GOTO 4470 ELSE GOTO 4440
                                                                        ' XDD
4360 IF QFCN%=4 THEN IF QPT%>=1 THEN GOSUB 6300 : GOTO 4470 ELSE GOTO 4440
                                                                        ' XSPL
4370 IF QFCN%=5 THEN IF QAM%>=0 THEN GOSUB 6400 : GOTO 4470 ELSE GOTO 4440
                                                                        ' XAM
4380 REM --- all others in MMIPART2 ---
4430 MDEX%=28 : GOSUB 7100                                              ' OOPS.
4440 MDEX%=31 : GOSUB 7100 : TM%=QFCN% : QFCN%=9 : D1%=0 : D2%=1 : D3%=0 : GOSUB 9200 : QFCN%=TM%
                                                                        ' Incomplete command - Deselect confirm.
4450 IF QFCN%>0 THEN D1%=0 : D2%=1 : D3%=1 : GOSUB 9200 : RETURN       ' Reselect function if one had been chosen.
4460 MDEX%=11 : GOSUB 7100 : RETURN                                     ' Ready to accept command.
4470 MDEX%=3 : GOSUB 7100 : QFCN%=9 : D1%=0 : D2%=1 : D3%=0 : GOSUB 9200 : QFCN%=0
                                                                        ' Completed command.
4490 IF GDTYPE%=1 GOTO 4460 ELSE IF LDE%<>0 AND SHUTDOWN%<>1 GOTO 4520
                                                                        ' Determine if we need to redisplay the DE.
4510 PRINT CHR$(27);"0A1";CHR$(12);CHR$(27);"0A3"; : GDTYPE%=-1 : GOTO 4460
                                                                        ' Erase graphics area.
4520 QMENU%=LDE% : GOSUB 6100 : GOTO 4460                              ' Redisplay the last DE.
4599 REM CANCEL ===========================================================
4600 QFCN%=10 : D1%=0 : D2%=1 : D3%=1 : GOSUB 9200 : MDEX%=2 : GOSUB 7100
                                                                        ' Backlight key.
4620 IF GDTYPE%=1 THEN 4630 ELSE 4650                                   ' Is there a DE on the screen.
4630 IF QPT%>0 THEN D1%=1 : D2%=2 : D3%=0 : GOSUB 9200 : GOTO 4670 ELSE GOTO 4670
                                                                        ' Deselect any selected point.
4640 PRINT CHR$(27);"0A1";CHR$(12);CHR$(27);"0A3"; : GDTYPE%=-1 : GOTO 4670
```

Main Program Segment
Annotated Source Code Listing

Energy Monitoring and Control System
Man/Machine Interface
Initial Release - May 1982

```
                                                                  ' Unable to redisplay last DE so erase graphics
                                                                    area.
4650 IF LDE%=0 GOTO 4640                                          ' Is it possible to redisplay the last DE.
4660 QMENU%=LDE% : DIAMEN%=LDE% : GOSUB 6100                      ' Redisplay the last DE.
4670 QFCN%=10 : QPT%=0 : QMENU%=0 : QAM%=-1 : ZC%=0 : QSCH%=0     ' Zero-out all selection variables.
4680 D1%=0 : D2%=1 : D3%=0 : GOSUB 9200 : QFCN%=0 : MDEX%=11 : GOSUB 7100 : RETURN
                                                                  ' Deselect cancel.
4699 REM MDIN ============================================================
4700 I%=1                                                         ' Loop till find which box selected.
4720 IF YTUCH%>=BOXLOC%(I%) GOTO 4740
4730 I%=I%+1 : IF I%>NBOXES%(DIAMEN%) THEN MDEX%=53 : GOSUB 7100 : RETURN ELSE GOTO 4720
4740 QMENU%=I% : D1%=-1 : D2%=2 : D3%=1 : GOSUB 9200 : MDEX%=18 : GOSUB 7100 : RETURN
                                                                  ' Highlight the selection.
4799 REM MLIN ============================================================
4800 I%=1                                                         ' Loop till find which box selected.
4820 IF YTUCH%>=BOXLOC%(I%) GOTO 4840
4830 I%=I%+1 : IF I%>NBOXES%(DIAMEN%) THEN MDEX%=53 : GOSUB 7100 : RETURN ELSE GOTO 4820
4840 D1%=-1 : D2%=2 : D3%=1 : QMENU%=I% : GOSUB 9200              ' Highlight the selection.
4850 MDEX%=10 : GOSUB 7100 : ZIN$="" : GOSUB 200                  ' Get new value from operator input.
4880 I%=1 : IF LEN(ZIN$)=0 GOTO 4910                              ' Did operator type in something.
4890 IF INSTR("0123456789",MID$(ZIN$,I%,1))=0 GOTO 4910           ' Make sure each character of input is a digit.
4900 I%=I%+1 : IF I%>LEN(ZIN$) GOTO 4920 ELSE GOTO 4890
4910 MDEX%=36 : GOSUB 7100 : GOTO 4850                            ' Typo error.
4920 IF QMENU%<>1 GOTO 4940                                       ' Is it a new low limit.
4930 IF VAL(ZIN$)<HTMP THEN LTMP=VAL(ZIN$) : GOTO 4990 ELSE MDEX%=32 : GOSUB 7100 : GOTO 4850
                                                                  ' Make sure it is in range before storing.
4940 IF QMENU%<>3 GOTO 4960                                       ' Is it a new high limit.
4950 IF VAL(ZIN$)>LTMP THEN HTMP=VAL(ZIN$) : GOTO 4990 ELSE MDEX%=29 : GOSUB 7100 : GOTO 4850
                                                                  ' Make sure it is in range before storing.
4960 IF QMENU%<>2 GOTO 4980                                       ' Is it a new set point.
4970 IF (VAL(ZIN$)>LTMP) AND (VAL(ZIN$)<HTMP) THEN STMP=VAL(ZIN$) : GOTO 4990 ELSE MDEX%=35 : GOSUB 7100 : GOTO 4850
                                                                  ' Make sure it is in range before storing.
4980 MDEX%=28 : GOSUB 7100                                        ' OOPS.
4990 D1%=-1 : D2%=2 : D3%=0 : GOSUB 9200 : MDEX%=23 : GOSUB 7100 : MDEX%=19 : GOSUB 7100 : RETURN
                                                                  ' Deselect menu item - ready for another.
5499 REM XSE =============================================================
5500 DSTAT%(QPT%)=0 : J%=0 : IF SSED%(QPT%)=1 THEN CDBVL(QPT%)=1   ' Enable and/or Start point.
5530 PRINT ""C2-Device ";PNAME$(ABS(PTYPE%(QPT%)),2);" Started/Enabled"
5540 D1%=1 : D2%=1 : D3%=2 : GOSUB 9200 : I%=1 : IF PTYPE%(QPT%)>0 GOTO 5680
                                                                  ' Redraw symbol.
5570 IF I%=QPT% GOTO 5660 ELSE TM%=I%                             '  Must also start/enable points tied to QPT% so
5590 IF PHIER%(TM%)=QPT% GOTO 5610                                ' Loop for this check.
```

Main Program Segment
Annotated Source Code Listing

Energy Monitoring and Control System
Man/Machine Interface
Initial Release - May 1982

```
5600 TM%=PHIER%(TM%) : IF TM%=0 GOTO 5660 ELSE GOTO 5590
5610 DSTAT%(I%)=0                                                ' Found one - so enable it.
5620 IF SSED%(I%)=1 THEN IF ABS(PTYPE%(QPT%))=4 THEN PRINT ""C2-Device :  ";PNAME$(ABS(PTYPE%(I%)),2);" Started" :
       CDBVL(I%)=1 : GOTO 5640 ELSE GOTO 5660                    ' also start it if applicable.
5630 PRINT ""C2-Device ";PNAME$(ABS(PTYPE%(I%)),2);" Enabled"
5640 IF LPNT%(I%)<>DIAMEN% GOTO 5660                             ' Is point on current DE.
5650 TP%=QPT% : QPT%=I% : D1%=1 : D2%=1 : D3%=2 : GOSUB 9200 : QPT%=TP%
                                                                 ' If so, redraw it.
5660 I%=I%+1 : IF I%<=NRTPTS% GOTO 5570                          ' Loop till checked entire database.
5670 IF QPT%=25 THEN J%=25 : QPT%=4 : DSTAT%(QPT%)=0 : CDBVL(QPT%)=1 : PRINT ""C2-Device :
       ";PN$(ABS(PTYPE%(QPT%)),2);" Started" : GOTO 5540         ' Special case - must also start pumping
                                                                   mechanism.
5671 IF QPT%=4 AND J%=25 THEN QPT%=25 : J%=0                     ' Return from special case.
5680 IF QPT%=APT% THEN GOSUB 9000                                ' If QPT% is current alarm then call ADISABLE.
5685 D1%=1 : D2%=2 : D3%=0 : GOSUB 9200                          ' Deselect point.
5690 QPT%=0 : RETURN                                             ' Done.
5799 REM XSD ==========================================================
5800 IF SSED%(QPT%)=1 THEN CDBVL(QPT%)=0 ELSE DSTAT%(QPT%)=1      ' Stop or disable point.
5830 PRINT ""C2-Device ";PNAME$(ABS(PTYPE%(QPT%)),2);" Stopped/Disabled"
5840 D1%=1 : D2%=1 : D3%=0 : GOSUB 9200 : IF PTYPE%(QPT%)>0 GOTO 6000 ' Redraw symbol.
5860 I%=1                                                        ' Must loop to also stop/disable points tied to
                                                                   QPT%.
5870 IF I%=QPT% GOTO 5980 ELSE TM%=I%                            ' Do not reprocess QPT%.
5890 IF PHIER%(TM%)=QPT% GOTO 5910                               ' Is point tied to QPT%.
5900 TM%=PHIER%(TM%) : IF TM%=0 GOTO 5980 ELSE GOTO 5890         ' Loop till found highest in path.
5910 IF SSED%(I%)=0 GOTO 5940                                    ' Found one - is it stopable.
5920 IF SSED%(QPT%)=0 GOTO 5980 ELSE CDBVL(I%)=0 : PRINT ""C2-Device ";PNAME$(ABS(PTYPE%(I%)),2);" Stopped" : GOTO 5960
                                                                 ' If initial command was to stop then also stop
                                                                   this point.
5940 DSTAT%(I%)=1 : PRINT ""C2-Device ";PNAME$(ABS(PTYPE%(I%)),2);", Disabled"
                                                                 ' Initial command was disable so disable this
                                                                   point.
5960 IF LPNT%(I%)<>DIAMEN% GOTO 5980                             ' Is point on current DE.
5970 TP%=QPT% : QPT%=I% : D1%=1 : D2%=1 : D3%=0 : GOSUB 9200 : QPT%=TP%
                                                                 ' Yes - so redraw it.
5980 I%=I%+1 : IF I%<=NRTPTS% GOTO 5870                          ' Loop till checked entire database.
6000 IF QPT%=APT% THEN GOSUB 9000                                ' If QPT% is current alarm then call ADISABLE.
6005 D1%=1 : D2%=2 : D3%=0 : GOSUB 9200                          ' Deselect point.
6010 QPT%=0 : RETURN                                             ' Done.
6099 REM XDD ==========================================================
6100 TM$="DIA"+MID$(STR$(QMENU%),2) : DOS"DRAW "+TM$              ' Get static DE image from disk.
6130 PRINT CHR$(27);"OA1";CHR$(12);CHR$(27);"W"; : I%=1           ' Display in graphics area.
```

Main Program Segment
Annotated Source Code Listing

Energy Monitoring and Control System
Man/Machine Interface
Initial Release - May 1982

```
6150 D1%=1 : D2%=3 : D3%=0 : IF LPNT%(I%)<>QMENU% GOTO 6210          ' Loop to draw symbols - check if on this DE.
6170 IF SSED%(I%)=1 THEN IF CDBVL(I%)=1 THEN D3%=D3%+2 : GOTO 6190 ELSE GOTO 6190
                                                                      ' Select color for start/stop.
6180 IF DSTAT%(I%)=0 THEN D3%=D3%+2                                   ' Select color for enable/disable.
6190 IF APT%=I% THEN D3%=D3%+8                                        ' Select flash or non-flash.
6200 QPT%=I% : GOSUB 9200 : QPT%=0                                    ' Draw symbol.
6210 I%=I%+1 : IF I%<=NRTPTS% GOTO 6150                               ' Loop till checked entire database.
6220 GDTYPE%=1 : DIAMEN%=QMENU% : QMENU%=0 : LDE%=DIAMEN% : IF QFCN%=3 THEN FDEAM%=1 : QFCN%=5 : D1%=0 : D2%=1 : D3%=0
     : GOSUB 9200 : QFCN%=0                                           ' Set up flags, choose auto mode if command was
                                                                        display diagram.
6240 PRINT CHR$(27);"OA3"; : RETURN                                   ' Return to text window - done.
6299 REM XSPL ==============================================================
6300 LOLIM(QPT%)=LTMP : HILIM(QPT%)=HTMP : SPT(QPT%)=STMP             ' Replace permanent with temporary storage.
6340 PRINT ""C2-New Set Point/Limits Accepted For Point:  ";PNAME$(ABS(PTYPE%(QPT%)),2)
6350 IF QPT%=APT% THEN GOSUB 9000                                     ' If point was current alarm then call ADISABLE.
6360 QPT%=0 : RETURN                                                  ' Done.
6399 REM XAM ===============================================================
6400 FDEAM%=QAM% : D1%=0 : D2%=1 : D3%=0 : GOSUB 9200                 ' Set flag and redraw key.
6440 PRINT ""C2-Mode Changed On DE:  ";TT$(1,DIAMEN%) : QA%=-1 : RETURN
                                                                      ' Done.
7099 REM DISMESS ===========================================================
7100 DOS "ACCESS 5"+STR$(MDEX%) : LINEINPUT #5;MESS$ : PRINT MESS$;""M"CO"2"N"C2" : RETURN
                                                                      ' Get message from disk and display it.
7199 REM DMENU =============================================================
7200 PRINT CHR$(27);"OA1";""M"CO"N";CHR$(12) : PRINT                  ' Erase graphics area.
7230 PRINT SPC((85-(LEN(TT$(DIAMEN%,0))*2))*2);""X2,"Y2,"C5";TT$(DIAMEN%,0);""X1,"Y1,";
                                                                      ' Display menu title.
7250 IF DIAMEN%=2 OR DIAMEN%=3 THEN LTMP=LOLIM(QPT%) : HTMP=HILIM(QPT%) : STMP=SPT(QPT%)
                                                                      ' If SET PT/LIM then set up temporary storage.
7270 D1%=-1 : D2%=3 : D3%=0 : FOR I%=1 TO NBOXES%(DIAMEN%) : QMENU%=I% : IF APT%<>0 AND QFCN%=3 THEN IF
     LPNT%(APT%)=QMENU% THEN D3%=8 ELSE D3%=0                         ' Loop for number of choices and select color.
7271 GOSUB 9200 : NEXT I%                                             ' Display each choice in correct color - loop
                                                                        till done.
7280 GDTYPE%=0 : QMENU%=0 : PRINT CHR$(27);"OA3"; : RETURN            ' Return to text window.
8999 REM ADISABLE ==========================================================
9000 GOSUB 110 : INC%=300 : GOSUB 160                                 ' Get current time plus 5 minutes.
9080 AQUE#(0,BQUE%)=APT% : AQUE#(1,BQUE%)=IJT# : BQUE%=BQUE%+1 : IF BQUE%>MQUE% THEN BQUE%=0
                                                                      ' Put alarm info in queue.
9090 IF BQUE%=TQUE% THEN PRINT ""C4"1ISYSTEM ERROR - Alarm Queue Overflow!"2" : MDEX%=28 : GOSUB 7100
                                                                      ' Check for overflow.
9115 D1%=3 : GOSUB 9200                                               ' Turn off alarm indicator.
9130 DSTAT%(APT%)=1 : SALARM%=0 : APT%=0 : RETURN                     ' Disable point and turn off alarm bell.
```

Main Program Segment
Annotated Source Code Listing

Energy Monitoring and Control System
Man/Machine Interface
Initial Release - May 1982

```
9149 REM Pre-DRAWIT Color Selection ------------------------------------------
9150 IF SSED%(QPT%)=1 THEN IF CDBVL(QPT%)=1 THEN D3%=D3%+2 ELSE D3%=D3%+0 ELSE IF DSTAT%(QPT%)=0 THEN D3%=D3%+2
                                                                      ' Select Point color.
9160 IF QPT%=APT% THEN D3%=D3%+8                                      ' Select blink if current alarm.
9170  GOTO 9200                                                       ' Ready to drawit.
9199 REM DRAWIT ==============================================================
9200 IF D1%<>-1 GOTO 9360                                             ' Is it a menu.
9220 REM MENU
9240 PRINT CHR$(27);"OA1^M^CO^N"; : IF (D3% AND 1)<>0 THEN PRINT "^C6^F^G+"; ELSE PRINT "^C3^F^G+";
9250 IF (D3% AND 8)<>0 THEN PRINT "^C5^1";
9260 PLOT 30,BOXLOC%(QMENU%),59,BOXLOC%(QMENU%)+29
9270 PRINT CHR$(21);"^L^K^U"; : PLOT 90,BOXLOC%(QMENU%)+20
9280 TM$=TT$(DIAMEN%,QMENU%) : PRINT TM$;
9300 IF QFCN%<>4 GOTO 9350
9310 IF QMENU%=1 THEN VTMP=LTMP ELSE IF QMENU%=2 THEN VTMP=STMP ELSE VTMP=HTMP
9340 PRINT USING "  ###.#";VTMP
9350 PRINT "^2";CHR$(27);"OA3"; : RETURN
9360 IF D1%<>0 GOTO 9660                                              ' Is it a function key.
9370 REM FKEY
9380 IF (D2% AND 1)=0 THEN RETURN
9400 PRINT CHR$(27);"OA2"; : IF (D3% AND 1)<>0 THEN PRINT "^C6"; ELSE PRINT "^C";RIGHT$(STR$(KCOLR%(1,QFCN%)),1)
9410 X1%=FKLOC%(QFCN%) : X2%=X1%+46 : Y1%=63 : Y2%=116
9420 PRINT "^F^G+"; : PLOT X1%,Y1%,X2%,Y2% : NLNS%=0
9440 FOR NKNT%=1 TO 3 : IF LEN(KTEXT$(NKNT%,QFCN%))<>0 THEN NLNS%=NLNS%+1
9450 NEXT NKNT%
9470 IF (QFCN%=5) AND (GDTYPE%=1) THEN NLNS%=1
9480 IF NLNS%=1 THEN Y1%=22 ELSE IF NLNS%=2 THEN Y1%=11 ELSE Y1%=6
9520 Y2%=116-Y1% : PRINT "^M^C";RIGHT$(STR$(KCOLR%(1,QFCN%)),1);"^N^C";RIGHT$(STR$(KCOLR%(2,QFCN%)),1)
9530 PRINT CHR$(21); : IF D3%=1 THEN PRINT "^M^C6^N^CO";
9540 FOR NKNT%=1 TO 3
9560 TM$=KTEXT$(NKNT%,QFCN%) : IF (QFCN%<>5) OR (GDTYPE%<>1) GOTO 9580
9570 IF FDEAM%=1 THEN TM$=LEFT$(KTEXT$(1,5),LEN(KTEXT$(1,5))-1) : PRINT "^C1"; ELSE TM$=KTEXT$(3,5) : PRINT "^C4";
9580 L%=LEN(TM$) : IF L%=0 GOTO 9640
9600 X2%=X1%+(47-(6*L%))/2
9610 PRINT "^K^U"; : PLOT X2%,Y2% : PRINT TM$
9620 IF NKNT%=NLNS% GOTO 9650
9630 Y2%=Y2%-Y1%-10
9640 NEXT  NKNT%
9650 PRINT CHR$(21);CHR$(27);"OA3"; : RETURN
9660 IF D1%<>1 GOTO 11000
9670 REM SYMBOL
9680 PQ%=ABS(PTYPE%(QPT%)) : SL%=LEN(SID$(PQ%))
```

Main Program Segment
Annotated Source Code Listing

Energy Monitoring and Control System
Man/Machine Interface
Initial Release - May 1982

```
9700 PRINT CHR$(27);"OA1"; : IF (D2% AND 2)=0 GOTO 10350
9710 REM BORDER
9720 X1%=PX%(QPT%) : Y1%=PY%(QPT%)
9730 IF PQ%<=2 THEN X2%=X1%+29 : Y2%=Y1%+29 : GOTO 9790
9740 IF PQ%<LIND% THEN X2%=X1%+21 : Y2%=Y1%+21 : GOTO 9790
9750 IF PQ%<17 THEN X2%=X1%+27 : Y2%=Y1%+17 : GOTO 9790
9760 IF PQ%=18 THEN X2%=X1%+45 : Y2%=Y1%+59 : GOTO 9790
9770 IF PQ%<>20 GOTO 9860
9780 X2%=X1%+75 : Y2%=Y1%+63
9790 IF PQ%<LIND% GOTO 9800 ELSE PRINT ""C1"; : GOTO 9810
9800 IF (D3% AND 1)=0 THEN PRINT ""C3"; ELSE PRINT ""C6";
9810 IF (D3% AND 8)<>0 THEN PRINT ""1";
9820 PRINT ""G("; : PLOT X1%,Y1%,X2%,Y1%,X2%,Y2%,X1%,Y2%,X1%,Y1% : PRINT ""2";
9830 IF D2%<>3 THEN PRINT CHR$(21);CHR$(27);"OA3"; : RETURN
9840 X1%=X1%+1 : X2%=X2%-1 : Y1%=Y1%+1 : Y2%=Y2%-1
9850 PRINT ""CO"G("; : PLOT  X1%,Y1%,X2%,Y1%,X2%,Y2%,X1%,Y2%,X1%,Y1% : PRINT CHR$(21) : GOTO 10350
9860 PRINT ""C1"; : IF PQ%<>19 GOTO 10000
9880 XC%=PX%(QPT%)+11 : YC%=PY%(QPT%)+11
9890 IF QPT%=84 GOTO 9950
9900 PRINT ""G)"; : PLOT XC%,YC%,11,90,270
9910 PRINT ""G("; : PLOT XC%+11,YC%,XC%+14,YC%,XC%+14,YC%+15,XC%-1,YC%+15
9920 PRINT ""CO"G)"; : PLOT XC%,YC%,10,90,270
9930 PRINT ""G("; : PLOT XC%+10,YC%,XC%+10,YC%+1,XC%+13,YC%+1,XC%+13,YC%+14,XC%-1,YC%+14
9940 IF D2%=3 GOTO 10350 ELSE PRINT CHR$(21);CHR$(27);"OA3"; : RETURN
9950 PRINT ""G)"; : PLOT XC%,YC%,11,180,270
9960 PRINT ""G("; : PLOT XC%-11,YC%,XC%-14,YC%,XC%-14,YC%+15,XC%,YC%+15
9970 PRINT ""CO"G)"; : PLOT XC%,YC%,10,180,270
9980 PRINT ""G("; : PLOT XC%-10,YC%,XC%-10,YC%+1,XC%-13,YC%+1,XC%-13,YC%+14,XC%,YC%+14
9990 IF D2%=3 GOTO 10350 ELSE PRINT CHR$(21);CHR$(27);"OA3"; : RETURN
10000 IF PQ%<>17 GOTO 10050
10010 XC%=PX%(QPT%)+11 : YC%=PY%(QPT%)+11
10020 PRINT ""G*"; : PLOT XC%,YC%,11
10030 PRINT ""CO"G*"; : PLOT XC%,YC%,10
10040 IF D2%=3 GOTO 10350 ELSE PRINT CHR$(21);CHR$(27);"OA3"; : RETURN
10050 IF PQ%<>21 GOTO 10180
10060 X1%=PX%(QPT%) : X2%=X1%+29 : X3%=X2%+6 : X4%=X1%-6 : Y1%=PY%(QPT%) : Y2%=Y1%+37 :
10070 PRINT ""G("; : PLOT X1%,Y1%,X2%,Y1%,X3%,Y2%,X4%,Y2%,X1%,Y1%
10080 X2%=X1%+8 : X3%=X2%+14 : Y2%=Y1%-8
10090 PRINT ""G("; : PLOT X2%,Y1%,X2%,Y2%,X3%,Y2%,X3%,Y1%
10100 X2%=X1%+2 : X3%=X2%+25 : X4%=X1%+13 : Y2%=Y1%+37 : Y3%=Y2%+2 : Y4%=Y3%+3
10110 PRINT ""G("; : PLOT X2%,Y2%,X2%,Y3%,X3%,Y3%,X3%,Y2%
10120 PRINT ""G("; : PLOT X2%,Y3%,X4%,Y4%,X3%,Y3%
```

Main Program Segment
Annotated Source Code Listing

Energy Monitoring and Control System
Man/Machine Interface
Initial Release - May 1982

```
10130 X1%=PX%(QPT%)+1 : X2%=X1%+27 : X3%=X2%+6 : X4%=X1%-6 : Y1%=PY%(QPT%)+1 : Y2%=Y1%+35
10140 PRINT "~C0~G("; : PLOT X1%,Y1%,X2%,Y1%,X3%,Y2%,X4%,Y2%,X1%,Y1%
10150 X2%=X1%+8 : X3%=X2%+12 : Y1%=Y1%-2 : Y2%=Y1%-6
10160 PRINT "~G("; : PLOT X2%,Y1%,X2%,Y2%,X3%,Y2%,X3%,Y1%,X2%,Y1%
10170 IF D2%=3 GOTO 10350 ELSE PRINT CHR$(21);CHR$(27);"OA3"; : RETURN
10180 IF PTYPE%(QPT%)<>22 GOTO 10300
10190 X1%=PX%(QPT%) : Y1%=PY%(QPT%)
10200 X2%=X1%+27 : X3%=X1%+5 : X4%=X2%-5 : X5%=X3%+9 : Y2%=Y1%+27 : Y3%=Y2%+2 : Y4%=Y3%+3
10210 PRINT "~G("; : PLOT X1%,Y1%,X2%,Y1%,X2%,Y2%,X1%,Y2%,X1%,Y1%
10220 PRINT "~G("; : PLOT X3%,Y2%,X3%,Y3%,X4%,Y3%,X4%,Y2%
10230 PRINT "~G("; : PLOT X3%,Y3%,X5%,Y4%,X4%,Y3%
10240 X1%=X1%+1 : X2%=X1%+25 : X3%=X1%+5 : X4%=X2%-5 : X5%=X3%+8 : Y1%=Y1%+1 : Y2%=Y1%+25 : Y3%=Y2%+2 : Y4%=Y3%+2 :
      Y5%=Y4%+1
10250 PRINT "~N~C0~G("; : PLOT X1%,Y1%,X2%,Y1%,X2%,Y2%,X1%,Y2%,X1%,Y1%
10260 PRINT "~G'"; : PLOT X3%,Y3%,X4%,Y3%
10270 X3%=X3%+5 : X4%=X4%-5
10280 PRINT "~G("; : PLOT X3%,Y4%,X5%,Y5%,X4%,Y4%,X3%,Y4%
10290 IF D2%=3 GOTO 10350 ELSE PRINT CHR$(21);CHR$(27);"OA3"; : RETURN
10300 REM COIL
10310 X1%=PX%(QPT%) : X2%=X1%+7 : Y1%=PY%(QPT%) : Y2%=Y1%+19
10320 PRINT "~G("; : PLOT X1%,Y1%,X2%,Y1%,X2%,Y2%,X1%,Y2%,X1%,Y1%
10330 PRINT "~C0~G("; : PLOT X1%+1,Y1%+1,X2%-1,Y1%+1,X2%-1,Y2%-1,X1%+1,Y2%-1,X1%+1,Y1%+1
10340 PRINT "~C1~G'"; : PLOT X1%,Y1%,X2%,Y2%
10350 REM INTERIOR
10360 IF (D2% AND 1)=0 THEN PRINT CHR$(21);CHR$(27);"OA3"; : RETURN
10370 X1%=PX%(QPT%)+2 : Y1%=PY%(QPT%)+2 : PQ%=ABS(PTYPE%(QPT%))
10380 IF PQ%<=2 THEN X2%=X1%+25 : Y2%=Y1%+25 : GOTO 10450
10390 IF PQ%>=LIND% GOTO 10400 ELSE X2%=X1%+17 : Y2%=Y1%+17 : GOTO 10450
10400 IF PQ%<17 THEN X2%=X1%+23 : Y2%=Y1%+13 : GOTO 10450
10410 IF PQ%=18 THEN X2%=X1%+41 : Y2%=Y1%+54 : GOTO 10450
10420 IF PQ%=20 THEN X2%=X1%+71 : Y2%=Y1%+58 : GOTO 10450
10430 IF PQ%=22 THEN X2%=X1%+23 : Y2%=Y1%+22 : GOTO 10450
10440 IF PQ%<>23 GOTO 10750 ELSE X2%=X1%+3 : Y2%=Y1%+15
10450 IF (D3% AND 2)<>0 THEN PRINT "~C2"; ELSE PRINT "~C0";
10460 PRINT "~F~G+"; : PLOT X1%,Y1%,X2%,Y2% : PRINT "~L";
10470 IF PQ%=23 THEN PRINT "~C1~G'"; : PLOT X1%,Y1%,X2%,Y2%
10490 PRINT CHR$(21) : IF SL%=0 THEN PRINT CHR$(27);"OA3"; : RETURN
10500 Y3%=Y2%-((Y2%-Y1%-9)/2) : L%=SL%
10510 IF (PQ%=1) OR (PQ%=2) OR (PQ%=18) THEN Y3%=Y2%-((Y2%-Y1%-19)/2) : L%=3
10520 IF PQ%=18 THEN L%=2
10530 X3%=X1%+(X2%-X1%-(6*L%+2))/2+1 : TM$=LEFT$(SID$(PQ%),L%)
10550 IF PQ%<LIND% THEN PRINT "~M~C0~N~C3"; : GOTO 10580
```

Main Program Segment
Annotated Source Code Listing

Energy Monitoring and Control System
Man/Machine Interface
Initial Release - May 1982

```
10570 PRINT "¯C1"; : IF (D3% AND 2)<>0 THEN PRINT "¯M¯C2¯N"; ELSE PRINT "¯M¯C0¯N";
10580 PRINT "¯K¯U"; : PLOT X3%,Y3% : PRINT TM$ : X3%=X3%+6*L%
10600 IF ((D3% AND 2)=0) OR (PQ%<LI%) THEN PRINT "¯C0"; ELSE PRINT "¯C2";
10610 PRINT "¯F¯G+"; : PLOT X3%,Y3%,X3%+1,Y3%-9 : PRINT "¯L";CHR$(21);"¯C3";
10630 IF (PQ%<>1) AND (PQ%<>2) THEN PRINT "¯C1"; : GOTO 10650
10640 L%=1 : GOTO 10660
10650 IF PQ%=18 THEN L%=6 ELSE GOTO 10740
10660 REM LINE2
10680 Y3%=Y3%-10 : X3%=X1%+(X2%-X1%-(6*L%+2))/2+1
10700 TM$=RIGHT$(SI$(PQ%),L%) : PRINT "¯K¯U"; : PLOT X3%,Y3% : PRINT TM$
10720 X3%=X3%+6*L% : IF ((D3% AND 2)=0) OR (PQ%<LI%) THEN PRINT "¯C0"; ELSE PRINT "¯C2";
10730 PRINT "¯F¯G+"; : PLOT X3%,Y3%,X3%+1,Y3%-9 : PRINT "¯L"; CHR$(21)
10740 PRINT CHR$(27);"OA3"; : RETURN
10750 REM ODD
10760 IF (D3% AND 2)<>0 THEN PRINT "¯C2"; ELSE PRINT "¯C0";
10770 IF PQ%=21 GOTO 10920
10790 X3%=X1%+9 : Y3%=Y1%+9 : IF PQ%=17 GOTO 10830
10800 REM PUMP
10810 IF QPT%<>84 THEN X4%=X3%+12 ELSE X4%=X3%-12
10820 PRINT "¯F¯G+"; : PLOT X3%,Y3%+2,X4%,Y3%+11
10830 REM PUMP/TC
10840 PRINT "¯F¯G*"; : PLOT X3%,Y3%,9 : PRINT "¯L";CHR$(21)
10860 IF PQ%<>17 THEN PRINT CHR$(27);"OA3"; : RETURN
10880 X4%=X3%-6 : Y4%=Y3%+5 : IF (D3% AND 2)<>0 THEN PRINT "¯M¯C2"; ELSE PRINT "¯M¯C0";
10890 PRINT "¯N¯C1¯K¯U"; : PLOT X4%,Y4% : PRINT SID$(PQ%)
10900 IF (D3% AND 2)<>0 THEN PRINT "¯C2"; ELSE PRINT "¯C0";
10910 PRINT "¯F¯G+"; : PLOT X4%+11,Y4%,X4%+12,Y4%-9 : PRINT "¯L";CHR$(21); CHR$(27);"OA3"; : RETURN
10920 REM TOWER
10930 X2%=X1%+25 : X3%=X2%+6 : X4%=X1%-6 : Y2%=Y1%+32
10940 IF (D3% AND 2)=0 THEN PRINT "¯C0¯1"; ELSE PRINT "¯M¯C2¯N¯C2";
10950 PRINT "¯G("; : PLOT X1%,Y1%,X2%,Y1%,X3%,Y2%,X4%,Y2%,X1%,Y1%
10960 IF (D3% AND 2)=0 THEN PRINT "¯M¯1¯C0¯N¯1¯C0";CHR$(14); ELSE PRINT "¯M¯C2¯2¯N¯C2";
10970 PRINT "¯K¯U"; : PLOT X1%+5,Y1%+5 : PRINT "¯>4";
10980 IF (D3% AND 2)=0 THEN PRINT "¯2¯M¯2¯C0¯ : 8";CHR$(12);"¯ : F¯C0¯N¯2";CHR$(15);
10990 PRINT "¯L";CHR$(21);CHR$(27);"OA3"; : RETURN
11000 REM ALARM
11020 PRINT CHR$(27);"OA1"; : X1%=5 : X2%=118 : Y1%=457 : Y2%=506
11030 IF D1%=3 THEN PRINT "¯C0"; ELSE PRINT "¯1¯C4";
11040 PRINT "¯F¯G+"; : PLOT X1%,Y1%,X2%,Y2% : PRINT "¯2¯L";CHR$(21);
11050 IF D1%=3 GOTO 11070
11060 PRINT "¯M¯1¯C4¯N¯2¯C7¯X3,¯Y4,¯K¯U"; : PLOT 17,501 : PRINT "ALARM";
11070 PRINT "¯X1,¯Y1,¯M¯2¯C0¯N";CHR$(21);CHR$(27);"OA3"; : RETURN
```

Secondary Program Segment
Annotated Source Code Listing

Energy Monitoring and Control System
Man/Machine Interface (EMCS/MMI)
Initial Release - May 1982

```
1 REM MMIPART2 - Secondary Program Segment
2 REM ======== (semi-condensed form -- annotated)
3 REM
4 REM GEORGIA INSTITUTE OF TECHNOLOGY
5 REM ENGINEERING EXPERIMENT STATION, COMMAND & CONTROL DIVISION
6 REM Development Team:  B.B.Wise, R.C.Coleburn, J.S.Cathcart, B.S.Rice
7 REM Initial Release - REV 1.00 - May 1982
8 REM
9 IF MC%=0 THEN DOS "CHAIN MMIINIT"                        ' Wrong sequence - run MMIINIT first.
10 MC%=2 : GOTO 250                                        ' Set flag that MMIPART2 now running.
30 STOP                                                    ' Fail-Safe.
69 REM KB ISR (Keyboard ISR) ==============================================
70 IF ERR=24 THEN T%=INP(&H4A) ELSE ON ERROR #0 GOTO 0
80 IF T%=8 THEN IF LEN(ZIN$)>0 THEN PRINT CHR$(27);"OA3";CHR$(8);" ";CHR$(8);CHR$(27);"OA0"; :
     ZIN$=LEFT$(ZIN$,LEN(ZIN$)-1) : RESUME
90 IF T%<>13 THEN IF T%<32ORT%>126 THEN RESUME ELSE PRINT CHR$(27);"OA3";CHR$(T%);CHR$(27);"OA0"; : ZIN$=ZIN$+CHR$(T%)
     : RESUME ELSE KISR%=1 : PRINT CHR$(27);"OA3"; : PRINT : PRINT CHR$(27);"OA0"; : RESUME
100 MDEX%=28 : GOSUB 7100 : STOP                           ' Fail-Safe.
109 REM GJTS ===============================================================
110 H%=PEEK(&H4321) : M%=PEEK(&H4322) : S%=PEEK(&H4323) : IF PEEK(&H4324)=1 THEN GOTO 120 ELSE
     CJT#=Y%*1000000000#+D%*1000000#+H%*10000#+M%*100#+S% : RETURN
120 D%=D%+1 : POKE &H4324,0 : IF Y%MOD4=0 THEN TP%=366 : M1%=1 ELSE TP%=365 : M1%=0
130 Y%=Y%+D%*TP% : D%=D%MODTP% : MT%=0 : IF Y%>99 THEN MDEX%=28 : GOSUB 7100
140 MT%=MT%+1 : IF D%>JDAY%(MT%,M1%) THEN 140
150 CJT#=Y%*1000000000#+D%*1000000#+H%*10000#+M%*100#+S% : IF MT%>1 THEN D%=M2%MODJDAY%(MT%-1,M1%) : RETURN ELSE
     M2%=D% : RETURN
159 REM IJTS ===============================================================
160 S9%=S%+IN% : M9%=M%+S9%*60 : S9%=S9%MOD60 : H9%=H%+M9%*60 : M9%=M9%MOD60 : D9%=D%+H9%*24 : H9%=H9%MOD24 : IF
     Y%MOD4=0 THEN TP%=366 ELSE TP%=365
170 Y9%=Y%+D%*TP% : D9%=D9%MODTP% : IF Y9%>99 THEN MDEX%=28 : GOSUB 7100
180 IJT#=Y9%*1000000000#+D9%*1000000#+H9%*10000#+M9%*100#+S9% : RETURN
190 REM EXEC ===============================================================
200 TISR%=0 : IQ%=1 : ON ERROR #1 GOTO 70  'Keyboard enable  - entry point.
201 IF HC%>0 GOTO 220 ELSE GOSUB 110                       ' Do not display time when help is running.
210 PRINT CHR$(27);"OA0";CHR$(28); : PRINT USING"**/";MT%,M2%; : PRINT USING"**  ";Y%; : PRINT USING"** : ";H%,M%; :
     PRINT USING"**";S%; : PRINT CHR$(27);"OA3";           ' Update time display.
220 IF BQUE%=TQUE% GOTO 240 ELSE IF AQUE#(1,TQUE%)>CJT# GOTO 240      ' Check alarm queue.
230 DSTAT%(AQUE#(0,TQUE%))=0 : TQUE%=TQUE%+1 : IF TQUE%=MQUE% THEN TQUE%=0
240 IF FDEAM%=1 AND CJT#>UC# AND IQ%<>1 THEN GOSUB 1800    ' Check if ready to update.
250 IF PEEK(&H42A0)=1 AND HC%=0 AND IQ%=0 THEN GOSUB 2100 : POKE&H42A0,0
                                                           ' Check for touch panel input.
```

Secondary Program Segment
Annotated Source Code Listing

Energy Monitoring and Control System
Man/Machine Interface (EMCS/MMI)
Initial Release - May 1982

```
260 IF KISR%=1 THEN IQ%=0 : KISR%=0 : ON ERROR #0 GOTO 0 : PRINT CHR$(27);"0A3"; : RETURN
                                                                       ' Check for keyboard input.
270 GOSUB 110 : IF SALARM%=1 AND PAUSEALRM#<=CJT# AND TC#<=CJT# THEN INC%=5 : GOSUB 160 : TC#=IJT# : PRINT CHR$(7);
                                                                       ' Check if ready to sound alarm.
280 IF SHUTDOWN%<>1 GOTO 201 ELSE ON ERROR #0 GOTO 0 : POKE&H4220,1    ' Check for shutdown.
290 DOS "DRAW SHUTDOWN" : PRINT CHR$(27);"0A1"M"C0"N";CHR$(12);CHR$(27);"W";CHR$(27);"0A3";
                                                                       ' Perform shutdown.
300 MDEX%=16 : GOSUB 7100 : NEW : END
1500 REM HELP ==============================================================
1510 HC%=1                                                             ' Set flag that help is running.
1530 H$="HLP"+MID$(STR$(HC%),2) : DOS "DRAW "+H$                       ' Retrieve help buffer file from disk.
1550 PRINT CHR$(12);CHR$(28);""="K";CHR$(27);"W"J";                    ' Display on screen.
1560 MDEX%=24 : GOSUB 7100 : ZIN$="" : GOSUB 200                       ' Wait for operator response.
1570 HC%=HC%+1                                                         ' Advance to next help screen.
1620 IF ZIN$<>"S" AND HC%<=NHELPBF% GOTO 1530                          ' Continue until last screen displayed or
                                                                         operator ready to stop.
1650 HC%=0 : DOS "DRAW MMISCRN" : PRINT CHR$(12);CHR$(28);""=";CHR$(27);"W";
                                                                       ' Turn off flag, redraw operator screen.
1660 PRINT ""W396502510492"M"2"C6"N"2"C0"P"K";CHR$(27);"0A1";           ' Define time window.
1670 PRINT ""W001130510451"M"2"C0"N"2"C1"P"K";CHR$(27);"0A2";           ' Define graphics window.
1680 PRINT ""W001051510128"M"2"C0"N"2"C1"P"K";CHR$(27);"0A3";           ' Define function key window.
1690 PRINT ""W001001510049"M"2"C0"N"2"C2"R"J";                          ' Define text window.
1700 TP%=QFCN% : QFCN%=5 : D1%=0 : D2%=1 : D3%=0 : GOSUB 9200 : QFCN%=TP% : GDTYPE%=-1 : RETURN
                                                                       ' Done.
1799 REM UPDATE ============================================================
1800 SIM%=SIM%+1 : IF SIM%>NNSIM% THEN SIM%=1                          ' Read in next simulation array.
1820 DOS "ARYLOAD SIM"+MID$(STR$(SIM%),2)+" NWVL" : TM%=1
1840 IF DSTAT%(TM%)=1 GOTO 1930                                        ' Ignore disabled points.
1850 IF NWVL(TM%)=CDBVL(TM%) GOTO 1930                                 ' Do not process points whose value has not
                                                                         changed.
1860 CDBVL(TM%)=NWVL(TM%) : IF SSED%(TM%)=1 AND GDTYPE%=1 AND DIAMEN%=LPNT%(TM%) THEN TP%=QPT% : QPT%=TM% : D1%=1 :
       D2%=3 : D3%=0 : GOSUB 9150 : QPT%=TP%                           ' Recolor point on screen if necessary.
1870 IF ABS(PTYPE%(TM%))<LALARM% OR ABS(PTYPE%(TM%))>=LALARM%+NALARM% GOTO 1930
                                                                       ' Can point generate an alarm.
1880 IF ADPT%(TM%)<>0 GOTO 1970                                        ' Analog or digital.
1890 IF (CDBVL(TM%)<LOLIM(TM%)) OR (CDBVL(TM%)>HILIM(TM%)) GOTO 1980   ' Check for analog alarm.
1900 IF ASTAT%(TM%)<>1 GOTO 1930                                       ' Was point previously in alarm.
1910 ASTAT%(TM%)=0 : ANPT%=TM% : GOSUB 2200                            ' Annunciate change.
1930 TM%=TM%+1 : IF TM%<=NRTPTS% GOTO 1840                             ' Loop for all points.
1950 DBRDY%=1 : GOSUB 110 : INC%=UT% : GOSUB 160 : UC#=IJT# : RETURN   ' Update time flag.
1970 IF CDBVL(TM%)<>ALARMVL%(TM%) GOTO 1900                            ' Check for digital alarm.
1980 IF ASTAT%(TM%)<>0 GOTO 1930                                       ' Was point previously not in alarm.
```

Secondary Program Segment
Annotated Source Code Listing

Energy Monitoring and Control System
Man/Machine Interface (EMCS/MMI)
Initial Release - May 1982

```
1990 ASTAT%(TM%)=1 : APT%=TM% : ANPT%=TM% : GOSUB 2200 : GOTO 1930      ' Annunciate change.
2100 REM TOUCHIN =========================================================
2105 CX%=PEEK(&H42A2) : CY%=PEEK(&H42A3)                                 ' Get touch coordinates from TPDRIVER.
2110 XTUCH%=FIX(CAL(1)+CAL(2)*CX%+CAL(3)*CX%^2+CAL(4)*CX%^3)             ' Calculate screen coordinates.
2111 YTUCH%=FIX(CAL(5)+CAL(6)*CY%+CAL(7)*CY%^2+CAL(8)*CY%^3)
2120 IF XTUCH%<0ORXTUCH%>511 THEN IF XTUCH%>511 THEN XTUCH%=511 ELSE XTUCH%=0
                                                                         ' Validate X.
2130 IF YTUCH%<0ORYTUCH%>511 THEN IF YTUCH%>511 THEN YTUCH%=511 ELSE YTUCH%=0
                                                                         ' Validate Y.
2140 IF YTUCH%<=116 AND YTUCH%>=63 THEN GOSUB 2400 : RETURN              ' Check for function key hit.
2150 IF YTUCH%>452 AND XTUCH%<=122 THEN GOSUB 3300 : RETURN              ' Check for alarm indicator hit.
2160 IF YTUCH%>452 OR YTUCH%<129 THEN RETURN                             ' Ignore invalid hits.
2170 IF GDTYPE%=1 THEN DOS "CHAIN MMIMAIN"                               ' Must go back to MMIMAIN to process symbol
                                                                           selection.
2180 IF GDTYPE%=0 THEN GOSUB 3100 : RETURN                               ' Menu Hit.
2190 DOS "CHAIN MMIMAIN"                                                 ' Default to MMIMAIN.
2199 REM ANNUNC ==========================================================
2200 D3%=0 : IF SSED%(ANPT%)=1 THEN IF CD(QN%)=1 THEN D3%=D3%+2 ELSE D3%=D3%+0 ELSE IF DSTAT%(ANPT%)=0 THEN D3%=D3%+2
                                                                         ' Determine point color.
2210 IF ASTAT%(ANPT%)=1 GOTO 2260                                        ' Is point the new alarm.
2220 PRINT ""M"C4"1"N"C7-Alarm Condition Corrected At:  ";PNAME$(ABS(PTYPE%(ANPT%)),2);""M"C0"2"N"
2230 IF GDTYPE%=1 AND DIAMEN%=LPNT%(ANPT%) THEN D1%=1 : D2%=3 : TP%=QPT% : QPT%=ANPT% : GOSUB 9200 : QPT%=ANPT%
                                                                         ' Redraw point if necessary.
2240 IF APT%=ANPT% THEN D1%=3 : GOSUB 9200 : SALARM%=0 : APT%=0          ' Turn off alarm indicator if necessary.
2250 RETURN
2260 D1%=2 : GOSUB 9200 : SALARM%=1 : PAUSEALRM#=0 : TC#=0               ' Turn on alarm indicator.
2280 PRINT ""M"C4"1"N"C7-ALarm Condition Detected At:  ";PNAME$(ABS(PTYPE%(ANPT%)),2);""M"C0"2"N"
2290 IF GDTYPE%=1 AND DIAMEN%=LPNT%(ANPT%) THEN D1%=1 : D2%=3 : D3%=D3%+8 : TP%=QPT% : QPT%=ANPT% : GOSUB 9200 :
     QPT%=ANPT% ELSE MDEX%=42 : GOSUB 7100                               ' Redraw point if necessary.
2300 RETURN
2400 REM FKEYIN ==========================================================
2410 IF QFCN%>0 THEN D1%=0 : D2%=1 : D3%=0 : GOSUB 9200 : IF XTUCH%<FKLOC%(9) THEN MDEX%=7 : GOSUB 7100 : QFCN%=0 :
     ZC%=0                                                               ' Deselect any previous function.
2460 IF XTUCH%<FKLOC%(6) THEN DOS "CHAIN MMIMAIN"                        ' Must go to MMIMAIN to process functions 1-5.
2470 IF XTUCH%<FKLOC%(7) THEN GOSUB 4000 : RETURN                        ' PRINT REPORT.
2480 IF XTUCH%<FKLOC%(8) THEN GOSUB 4100 : RETURN                        ' MODIFY SCHED.
2490 IF XTUCH%<FKLOC%(9) THEN GOSUB 4200 : RETURN                        ' CHANGE OPER.
2500 IF XTUCH%<FKLOC%(10) THEN GOSUB 4300 : RETURN                       ' CONFIRM.
2510 GOSUB 4600 : RETURN                                                 ' CANCEL.
3100 REM MENUIN ==========================================================
3110 IF ZC%=1 THEN GOSUB 11300 : RETURN                                  ' IF MODIFY SCHED running then goto MSCHED.
3120 IF QMENU%=0 GOTO 3180 ELSE D1%=-1 : D2%=1 : D3%=0                   ' Deselect any previous menu selection.
```

Secondary Program Segment
Annotated Source Code Listing

Energy Monitoring and Control System
Man/Machine Interface (EMCS/MMI)
Initial Release - May 1982

```
3130 IF DIAMEN%=6 THEN IF RODS%>0 THEN TP%=QMENU% : QMENU%=RODS% : GOSUB 9200 : QMENU%=TP% : RODS%=0 : GOTO 3150 ELSE
     GOTO 3180
3140 GOSUB 9200 : QMENU%=0
3150 MDEX%=8 : GOSUB 7100
3180 IF DIAMEN%=4 THEN GOSUB 5200 : RETURN                               ' MRIN.
3190 IF DIAMEN%=5 THEN GOSUB 5100 : RETURN                               ' MOIN.
3200 IF DIAMEN%=6 THEN GOSUB 5400 : RETURN                               ' REPOUT.
3210 IF DIAMEN%=7 THEN GOSUB 5300 : RETURN                               ' MSIN.
3220 MDEX%=28 : GOSUB 7100 : RETURN                                      ' OOPS.
3300 GOSUB 110 : INC%=120 : GOSUB 160 : PAUSEALRM#=IJT# : TC#=IJT# : MDEX%=39 : GOSUB 7100 : RETURN
                                                                         ' Temporarily silence alarm bell.
4000 REM REPORT ==============================================================
4010 QFCN%=6 : D1%=0 : D2%=1 : D3%=1 : GOSUB 9200 : QPT%=0 : QMENU%=0 : DIAMEN%=4 : RODS%=0 : GOSUB 7200 : MDEX%=22 :
     GOSUB 7100 : RETURN                                                 ' Backlight key and draw menu.
4100 REM SCHED ===============================================================
4110 QFCN%=7 : D1%=0 : D2%=1 : D3%=1 : GOSUB 9200 : QSCH%=0 : QMENU%=0 : DIAMEN%=7 : GOSUB 7200 : MDEX%=48 : GOSUB
     7100 : RETURN                                                       ' Backlight key and draw menu.
4200 REM OPER ================================================================
4210 QFCN%=8 : D1%=0 : D2%=1 : D3%=1 : GOSUB 9200 : QPT%=0 : QMENU%=0 : QAM%=0 : DIAMEN%=5 : QOPER%=0 : MDEX%=12 :
     GOSUB 7100 : GOSUB 7200 : MDEX%=43 : GOSUB 7100 : RETURN            ' Backlight key and draw menu.
4300 REM CONFIRM =============================================================
4310 TM%=QFCN% : QFCN%=9 : D1%=0 : D2%=1 : D3%=1 : GOSUB 9200 : QFCN%=TM%
                                                                         ' Backlight key.
4320 IF (QFCN%<=0) OR (QFCN%>8) THEN 4440                                ' Make sure a function key has been selected.
4380 IF QFCN%=6 AND RODS%=0 AND QMENU%<=NU% AND QMENU%>=1 THEN DIAMEN%=6 : TM%=QMENU% : GOSUB 7200 : QMENU%=TM% :
     TM%=QFCN% : QFCN%=9 : D1%=0 : D2%=1 : D3%=0 : GOSUB 9200 : QFCN%=TM% : MDEX%=47 : GOSUB 7100 : RETURN
                                                                         ' PRINT REPORT - Draw 2nd menu level
4390 IF QFCN%=6 THEN IF QMENU%>=1 AND QMENU%<=NUMRPTS% AND RODS%>=1 THEN GOSUB 6500 : GOTO 4470 :  ELSE GOTO 4440
                                                                         ' Have 2nd menu, now call XPR.
4400 IF QFCN%=7 AND ZC%=0 THEN GOSUB 11100 : ZC%=1 : RETURN              ' MODIFY SCHED - Call DSCHED.
4410 IF QFCN%=7 THEN IF QSCH%>=1 AND QSCH%<=NDES% THEN GOSUB 6700 : GOTO 4470 :  ELSE GOTO 4440
                                                                         ' MODIFY SCHED - Have done MSCHED, now call XMS.
4420 IF QFCN%=8 THEN IF QMENU%>=1 AND QMENU%<=3 THEN GOSUB 6800 : GOTO 4470 ELSE GOTO 4440
                                                                         ' XCHOP.
4430 MDEX%=28 : GOSUB 7100                                               ' OOPS.
4440 MDEX%=31 : GOSUB 7100 : TM%=QFCN% : QFCN%=9 : D1%=0 : D2%=1 : D3%=0 : GOSUB 9200 : QFCN%=TM%
                                                                         ' Incomplete command, Deselect confirm.
4450 IF QFCN%>0 THEN D1%=0 : D2%=1 : D3%=1 : GOSUB 9200 : RETURN         ' Reselect function if one had been chosen.
4460 MDEX%=11 : GOSUB 7100 : RETURN                                      ' Ready to accept command.
4470 MDEX%=3 : GOSUB 7100 : QFCN%=9 : D1%=0 : D2%=1 : D3%=0 : GOSUB 9200 : QFCN%=0
                                                                         ' Completed command.
4490 IF GDTYPE%=1 GOTO 4460 ELSE IF LDE%<>0 AND SHUTDOWN%<>1 GOTO 4520
```

Secondary Program Segment
Annotated Source Code Listing

Energy Monitoring and Control System
Man/Machine Interface (EMCS/MMI)
Initial Release - May 1982

```
                                                                  ' Determine if DE needs to be redisplayed.
4510 PRINT CHR$(27);"OA1";CHR$(12);CHR$(27);"OA3"; : GDTYPE%=-1 : GOTO 4460
                                                                  ' Erase graphics area.
4520 QMENU%=LDE% : GOSUB 6100 : GOTO 4460                         ' Redisplay last DE.
4600 REM CANCEL =========================================================
4610 QFCN%=10 : D1%=0 : D2%=1 : D3%=1 : GOSUB 9200 : MDEX%=2 : GOSUB 7100
                                                                  ' Backlight key.
4620 IF GDTYPE%=1 THEN 4630 ELSE 4650                             ' Is there a DE presently displayed.
4630 IF QPT%>0 THEN D1%=1 : D2%=2 : D3%=0 : GOSUB 9200 : GOTO 4670 ELSE GOTO 4670
                                                                  ' Deselect any selected point.
4640 PRINT CHR$(27);"OA1";CHR$(12);CHR$(27);"OA3"; : GDTYPE%=-1 : GOTO 4670
                                                                  ' Unable to redisplay last DE, so erase graphics
                                                                    area.
4650 IF LDE%=0 GOTO 4640                                          ' Is it possible to redisplay the last DE.
4660 QMENU%=LDE% : DIAMEN%=LDE% : GOSUB 6100                      ' Redisplay the last DE.
4670 QFCN%=10 : QPT%=0 : QMENU%=0 : QAM%=-1 : ZC%=0 : QSCH%=0     ' Zero-out all selection variables.
4680 D1%=0 : D2%=1 : D3%=0 : GOSUB 9200 : QFCN%=0 : MDEX%=11 : GOSUB 7100 : RETURN
                                                                  ' Deselect any selected funtion.
5100 REM MOIN ===========================================================
5110 I%=1                                                         ' Loop till found which box was selected.
5120 IF YTUCH%>=BOXLOC%(I%) GOTO 5140
5130 I%=I%+1 : IF I%>NBOXES%(DIAMEN%) THEN MDEX%=53 : GOSUB 7100 : RETURN ELSE GOTO 5120
5140 QMENU%=I% : D1%=-1 : D2%=2 : D3%=1 : GOSUB 9200 : MDEX%=18 : GOSUB 7100 : RETURN
                                                                  ' Highlight selection.
5200 REM MRIN ===========================================================
5210 I%=1                                                         ' Loop till found which box was selected.
5220 IF YTUCH%>=BOXLOC%(I%) GOTO 5240
5230 I%=I%+1 : IF I%>NBOXES%(DIAMEN%) THEN MDEX%=53 : GOSUB 7100 : RETURN ELSE GOTO 5220
5240 QMENU%=I% : D1%=-1 : D2%=2 : D3%=1 : GOSUB 9200 : MDEX%=9 : GOSUB 7100 : RETURN
                                                                  ' Highlight selection.
5300 REM MSIN ===========================================================
5310 I%=1                                                         ' Loop till found which box was selected.
5320 IF YTUCH%>=BOXLOC%(I%) GOTO 5340
5330 I%=I%+1 : IF I%>NBOXES%(DIAMEN%) THEN MDEX%=53 : GOSUB 7100 : RETURN ELSE GOTO 5320
5340 D1%=-1 : D2%=2 : D3%=1 : QMENU%=I% : GOSUB 9200 : QSCH%=QMENU% : MDEX%=51 : GOSUB 7100 : RETURN
                                                                  ' Highlight selection.
5400 REM REPOUT =========================================================
5410 I%=1                                                         ' Loop till found which box was selected.
5420 IF YTUCH%>=BOXLOC%(I%) GOTO 5440
5430 I%=I%+1 : IF I%>NBOXES%(DIAMEN%) THEN MDEX%=53 : GOSUB 7100 : RETURN ELSE GOTO 5420
5440 RODS%=I% : TP%=QMENU% : QMENU%=RODS% : D1%=-1 : D2%=1 : D3%=1 : GOSUB 9200 : QMENU%=TP% : MDEX%=18 : GOSUB 7100 :
       RETURN                                                     ' Highlight selection.
```

Secondary Program Segment
Annotated Source Code Listing

Energy Monitoring and Control System
Man/Machine Interface (EMCS/MMI)
Initial Release - May 1982

```
6100 REM XDD ===============================================================
6110 TM$="DIA"+MID$(STR$(QMENU%),2) : DOS "DRAW "+TM$            ' Get static DE image from disk.
6130 PRINT CHR$(27);"OA1";CHR$(12);CHR$(27);"W"; : I%=1          ' Display in graphics area.
6150 D1%=1 : D2%=3 : D3%=0 : IF LPNT%(I%)<>QMENU% GOTO 6210      ' Loop to draw symbols - check if on this DE.
6170 IF SSED%(I%)=1 THEN IF CDBVL(I%)=1 THEN D3%=D3%+2 : GOTO 6190 ELSE GOTO 6190
                                                                 ' Select color for start/stop.
6180 IF DSTAT%(I%)=0 THEN D3%=D3%+2                              ' Select color for enable disable.
6190 IF APT%=I% THEN D3%=D3%+8                                   ' Select flashing or non-flashing.
6200 QPT%=I% : GOSUB 9200 : QPT%=0                               ' Draw symbol.
6210 I%=I%+1 : IF I%<=NRTPTS% GOTO 6150                          ' Loop till checked entire database.
6220 GDTYPE%=1 : DIAMEN%=QMENU% : QMENU%=0 : LDE%=DIAMEN% : IF QFCN%=3 THEN FDEAM%=1 : QFCN%=5 : D1%=0 : D2%=1 : D3%=0
    : GOSUB 9200 : QFCN%=0                                       ' Set up flags, choose auto mode if function is
                                                                   display diagram.
6240 PRINT CHR$(27);"OA3"; : RETURN                              ' Return to text window - Done.
6500 REM XPR ===============================================================
6550 TM$="RPT"+MID$(STR$(QMENU%),2,1) : DOS "DRAW "+TM$          ' Get static report from disk.
6570 IF RODS%=1 GOTO 6620                                        ' Display on printer or crt.
6580 PRINT CHR$(27);"OA1"M"C0"N"C7"R";CHR$(12);CHR$(27);"W";CHR$(27);"OA3";
                                                                 ' Display report in graphics area.
6595 MDEX%=54 : GOSUB 7100 : ZIN$="" : GOSUB 200                 ' Wait for operator response.
6600 GDTYPE%=-1 : QMENU%=0 : RODS%=0 : RETURN                    ' Done.
6620 PRINT CHR$(27);"R07";CHR$(27);"U"; : LPRINT CHR$(12);CHR$(13); : GOTO 6600
                                                                 ' Display report on printer.
6700 REM XMS ===============================================================
6710 FOR I%=1 TO NSCH% : S$(QSCH%,I%)=SCHTMP$(I%) : NEXT I%      ' Replace permanent with temporary storage.
6720 PRINT ""C2-New Schedule Accepted For DE: ";TT$(1,QSCH%)
6730 QSCH%=0 : ZC%=0 : QMENU%=0 : RETURN                         ' Done.
6800 REM XCHOP =============================================================
6810 IF QMENU%=2 THEN GOSUB 1500 : RETURN                        ' HELP.
6820 IF QMENU%=3 THEN SHUTDOWN%=1 : RETURN                       ' SHUTDOWN.
6830 MDEX%=13 : GOSUB 7100 : ZIN$="" : GOSUB 200                 ' CHANGE OPERATOR - Get name.
6840 I%=1
6870 IF ZIN$=OPER$(I%) GOTO 6900                                 ' Validate operator.
6890 I%=I%+1 : IF I%>NOPER% THEN MDEX%=33 : GOSUB 7100 : GOTO 7000 ELSE 6870
6900 MDEX%=4 : GOSUB 7100 : ZIN$="" : GOSUB 200                  ' Get password
6930 IF ZIN$=PWRD$(I%) GOTO 6950 ELSE MDEX%=34 : GOSUB 7100 : GOTO 7000
                                                                 ' Validate password.
6950 PRINT ""C2-Password Accepted.  New Operator Is ";OPER$(I%);""P"; : CX=CURSX(3) : CY=CURSY(3)
                                                                 ' Save cursor position.
6970 PRINT ""U396485"M"C6"2"N"C0"2";OPER$(I%);""3"M"C0"N"C2"U"; : PLOT CX,CY : PRINT ""R"; : PRINT  : QOPER%=I%
                                                                 ' Display new operator name in DTO window and
                                                                   return to previous cursor position.
```

Secondary Program Segment
Annotated Source Code Listing

Energy Monitoring and Control System
Man/Machine Interface (EMCS/MMI)
Initial Release - May 1982

```
7000 RETURN                                                               ' Done.
7100 REM DISMESS ==========================================================
7110 DOS "ACCESS 5"+STR$(MDEX%) : LINEINPUT #5;MESS$ : PRINT MESS$;""M"C0"2"N"C2" : RETURN
                                                                          ' Retrieve message from disk and display in text
                                                                            area.
7200 REM DMENU ============================================================
7210 PRINT CHR$(27);"0A1";""M"C0"N";CHR$(12) : PRINT                      ' Erase graphics area.
7230 PRINT SPC((85-(LEN(TT$(DIAMEN%,0))*2))*2);""X2,"Y2,"C5";TT$(DIAMEN%,0);""X1,"Y1,";
                                                                          ' Display menu title.
7250 IF DIAMEN%=2 OR DIAMEN%=3 THEN LTMP=LOLIM(QPT%) : HTMP=HILIM(QPT%) : STMP=SPT(QPT%)
                                                                          ' IF SETPT/LIMITS then set up temporary storage.
7270 D1%=-1 : D2%=3 : D3%=0 : FOR I%=1 TO NBOXES%(DIAMEN%) : QMENU%=I% : IF APT%>0 AND QFCN%=3 THEN IF
     LPNT%(APT%)=QMENU% THEN D3%=8 ELSE D3%=0                             ' Loop for number of choices and select color.
7271 GOSUB 9200 : NEXT I%                                                 ' Display each choice in correct color - loop
                                                                            till done.
7280 GDTYPE%=0 : QMENU%=0 : PRINT CHR$(27);"0A3"; : RETURN                ' Return to text window.
9149 REM Pre-Drawit color selection ---------------------------------------
9150 IF SSED%(QPT%)=1 THEN IF CD(QPT%)=1 THEN D3%=D3%+2 ELSE D3%=D3%+0 ELSE IF DSTAT%(QPT%)=0 THEN D3%=D3%+2
                                                                          ' Select point interior color.
9160 IF QPT%=APT% THEN D3%=D3%+8                                          ' Select flashing if current alarm.
9170 GOTO 9200                                                            ' Ready to drawit.
9200 REM DRAWIT ===========================================================
9210 IF D1%<>-1 GOTO 9360
9220 REM MENU
9240 PRINT CHR$(27);"0A1"M"C0"N"; : IF (D3% AND 1)<>0 THEN PRINT ""C6"F"G+"; ELSE PRINT ""C3"F"G+";
9250 IF (D3% AND 8)<>0 THEN PRINT ""C5"1";
9260 PLOT 30,BOXLOC%(QMENU%),59,BOXLOC%(QMENU%)+29
9270 PRINT CHR$(21);""L"K"U"; : PLOT 90,BOXLOC%(QMENU%)+20
9280 TM$=TT$(DIAMEN%,QMENU%) : PRINT TM$;
9300 IF QFCN%<>4 GOTO 9350
9310 IF QMENU%=1 THEN VTMP=LTMP ELSE IF QMENU%=2 THEN VTMP=STMP ELSE VTMP=HTMP
9340 PRINT USING "  ###.#";VTMP
9350 PRINT ""2";CHR$(27);"0A3"; : RETURN
9360 IF D1%<>0 GOTO 9660
9370 REM FKEY
9380 IF (D2% AND 1)=0 THEN RETURN
9400 PRINT CHR$(27);"0A2"; : IF (D3% AND 1)<>0 THEN PRINT ""C6"; ELSE PRINT ""C";RIGHT$(STR$(KCOLR%(1,QFCN%)),1)
9410 X1%=FKLOC%(QFCN%) : X2%=X1%+46 : Y1%=63 : Y2%=116
9420 PRINT ""F"G+"; : PLOT X1%,Y1%,X2%,Y2% : NLNS%=0
9440 FOR NKNT%=1TO3 : IF LEN(KTEXT$(NKNT%,QFCN%))<>0 THEN NLNS%=NLNS%+1
9450 NEXT NKNT%
9470 IF (QFCN%=5) AND (GDTYPE%=1) THEN NLNS%=1
```

Secondary Program Segment
Annotated Source Code Listing

Energy Monitoring and Control System
Man/Machine Interface (EMCS/MMI)
Initial Release - May 1982

```
9480 IF NLNS%=1 THEN Y1%=22 ELSE IF NLNS%=2 THEN Y1%=11 ELSE Y1%=6
9520 Y2%=116-Y1% : PRINT ""M"C";RIGHT$(STR$(KCOLR%(1,QFCN%)),1);""N"C";RIGHT$(STR$(KCOLR%(2,QFCN%)),1)
9530 PRINT CHR$(21); : IF D3%=1 THEN PRINT ""M"C6"N"C0";
9540 FOR NKNT%=1TO3
9560 TM$=KTEXT$(NKNT%,QFCN%) : IF (QFCN%<>5) OR (GDTYPE%<>1) GOTO 9580
9570 IF FDEAM%=1 THEN TM$=LEFT$(KTEXT$(1,5),LEN(KTEXT$(1,5))-1) : PRINT ""C1"; ELSE TM$=KTEXT$(3,5) : PRINT ""C4";
9580 L%=LEN(TM$) : IF L%=0 GOTO 9640
9600 X2%=X1%+(47-(6*L%))/2
9610 PRINT ""K"U"; : PLOT X2%,Y2% : PRINT TM$
9620 IF NKNT%=NLNS% GOTO 9650
9630 Y2%=Y2%-Y1%-10
9640 NEXT NKNT%
9650 PRINT CHR$(21);CHR$(27);"OA3"; : RETURN
9660 IF D1%<>1 GOTO 11000
9670 REM SYMBOL
9680 PQ%=ABS(PTYPE%(QPT%)) : SL%=LEN(SID$(PQ%))
9700 PRINT CHR$(27);"OA1"; : IF (D2% AND 2)=0 GOTO 10350
9710 REM BORDER
9720 X1%=PX%(QPT%) : Y1%=PY%(QPT%)
9730 IF PQ%<=2 THEN X2%=X1%+29 : Y2%=Y1%+29 : GOTO 9790
9740 IF PQ%<LIND% THEN X2%=X1%+21 : Y2%=Y1%+21 : GOTO 9790
9750 IF PQ%<17 THEN X2%=X1%+27 : Y2%=Y1%+17 : GOTO 9790
9760 IF PQ%=18 THEN X2%=X1%+45 : Y2%=Y1%+59 : GOTO 9790
9770 IF PQ%<>20 GOTO 9860
9780 X2%=X1%+75 : Y2%=Y1%+63
9790 IF PQ%<LIND% GOTO 9800 ELSE PRINT ""C1"; : GOTO 9810
9800 IF (D3% AND 1)=0 THEN PRINT ""C3"; ELSE PRINT ""C6";
9810 IF (D3% AND 8)<>0 THEN PRINT ""1";
9820 PRINT ""G("; : PLOT X1%,Y1%,X2%,Y1%,X2%,Y2%,X1%,Y2%,X1%,Y1% : PRINT ""2";
9830 IF D2%<>3 THEN PRINT CHR$(21);CHR$(27);"OA3"; : RETURN
9840 X1%=X1%+1 : X2%=X2%-1 : Y1%=Y1%+1 : Y2%=Y2%-1
9850 PRINT ""C0"G("; : PLOT  X1%,Y1%,X2%,Y1%,X2%,Y2%,X1%,Y2%,X1%,Y1% : PRINT CHR$(21) : GOTO 10350
9860 PRINT ""C1"; : IF PQ%<>19 GOTO 10000
9880 XC%=PX%(QPT%)+11 : YC%=PY%(QPT%)+11
9890 IF QPT%=84 GOTO 9950
9900 PRINT ""G)"; : PLOT XC%,YC%,11,90,270
9910 PRINT ""G("; : PLOT XC%+11,YC%,XC%+14,YC%,XC%+14,YC%+15,XC%-1,YC%+15
9920 PRINT ""C0"G)"; : PLOT XC%,YC%,10,90,270
9930 PRINT ""G("; : PLOT XC%+10,YC%,XC%+10,YC%+1,XC%+13,YC%+1,XC%+13,YC%+14,XC%-1,YC%+14
9940 IF D2%=3 GOTO 10350 ELSE PRINT CHR$(21);CHR$(27);"OA3"; : RETURN
9950 PRINT ""G)"; : PLOT XC%,YC%,11,180,270
9960 PRINT ""G("; : PLOT XC%-11,YC%,XC%-14,YC%,XC%-14,YC%+15,XC%,YC%+15
```

Secondary Program Segment
Annotated Source Code Listing

Energy Monitoring and Control System
Man/Machine Interface (EMCS/MMI)
Initial Release - May 1982

```
9970 PRINT "^CO^G)"; : PLOT XC%,YC%,10,180,270
9980 PRINT "^G("; : PLOT XC%-10,YC%,XC%-10,YC%+1,XC%-13,YC%+1,XC%-13,YC%+14,XC%,YC%+14
9990 IF D2%=3 GOTO 10350 ELSE PRINT CHR$(21);CHR$(27);"OA3"; : RETURN
10000 IF PQ%<>17 GOTO 10050
10010 XC%=PX%(QPT%)+11 : YC%=PY%(QPT%)+11
10020 PRINT "^G*"; : PLOT XC%,YC%,11
10030 PRINT "^CO^G*"; : PLOT XC%,YC%,10
10040 IF D2%=3 GOTO 10350 ELSE PRINT CHR$(21);CHR$(27);"OA3"; : RETURN
10050 IF PQ%<>21 GOTO 10180
10060 X1%=PX%(QPT%) : X2%=X1%+29 : X3%=X2%+6 : X4%=X1%-6 : Y1%=PY%(QPT%) : Y2%=Y1%+37 :
10070 PRINT "^G("; : PLOT X1%,Y1%,X2%,Y1%,X3%,Y2%,X4%,Y2%,X1%,Y1%
10080 X2%=X1%+8 : X3%=X2%+14 : Y2%=Y1%-8
10090 PRINT "^G("; : PLOT X2%,Y1%,X2%,Y2%,X3%,Y2%,X3%,Y1%
10100 X2%=X1%+2 : X3%=X2%+25 : X4%=X1%+13 : Y2%=Y1%+37 : Y3%=Y2%+2 : Y4%=Y3%+3
10110 PRINT "^G("; : PLOT X2%,Y2%,X2%,Y3%,X3%,Y3%,X3%,Y2%
10120 PRINT "^G("; : PLOT X2%,Y3%,X4%,Y4%,X3%,Y3%
10130 X1%=PX%(QPT%)+1 : X2%=X1%+27 : X3%=X2%+6 : X4%=X1%-6 : Y1%=PY%(QPT%)+1 : Y2%=Y1%+35
10140 PRINT "^CO^G("; : PLOT X1%,Y1%,X2%,Y1%,X3%,Y2%,X4%,Y2%,X1%,Y1%
10150 X2%=X1%+8 : X3%=X2%+12 : Y1%=Y1%-2 : Y2%=Y1%-6
10160 PRINT "^G("; : PLOT X2%,Y1%,X2%,Y2%,X3%,Y2%,X3%,Y1%,X2%,Y1%
10170 IF D2%=3 GOTO 10350 ELSE PRINT CHR$(21);CHR$(27);"OA3"; : RETURN
10180 IF PTYPE%(QPT%)<>22 GOTO 10300
10190 X1%=PX%(QPT%) : Y1%=PY%(QPT%)
10200 X2%=X1%+27 : X3%=X1%+5 : X4%=X2%-5 : X5%=X3%+9 : Y2%=Y1%+27 : Y3%=Y2%+2 : Y4%=Y3%+3
10210 PRINT "^G("; : PLOT X1%,Y1%,X2%,Y1%,X2%,Y2%,X1%,Y2%,X1%,Y1%
10220 PRINT "^G("; : PLOT X3%,Y2%,X3%,Y3%,X4%,Y3%,X4%,Y2%
10230 PRINT "^G("; : PLOT X3%,Y3%,X5%,Y4%,X4%,Y3%
10240 X1%=X1%+1 : X2%=X1%+25 : X3%=X1%+5 : X4%=X2%-5 : X5%=X3%+8 : Y1%=Y1%+1 : Y2%=Y1%+25 : Y3%=Y2%+2 : Y4%=Y3%+2 :
      Y5%=Y4%+1
10250 PRINT "^N^CO^G("; : PLOT X1%,Y1%,X2%,Y1%,X2%,Y2%,X1%,Y2%,X1%,Y1%
10260 PRINT "^G!"; : PLOT X3%,Y3%,X4%,Y3%
10270 X3%=X3%+5 : X4%=X4%-5
10280 PRINT "^G("; : PLOT X3%,Y4%,X5%,Y5%,X4%,Y4%,X3%,Y4%
10290 IF D2%=3 GOTO 10350 ELSE PRINT CHR$(21);CHR$(27);"OA3"; : RETURN
10300 REM COIL
10310 X1%=PX%(QPT%) : X2%=X1%+7 : Y1%=PY%(QPT%) : Y2%=Y1%+19
10320 PRINT "^G("; : PLOT X1%,Y1%,X2%,Y1%,X2%,Y2%,X1%,Y2%,X1%,Y1%
10330 PRINT "^CO^G("; : PLOT X1%+1,Y1%+1,X2%-1,Y1%+1,X2%-1,Y2%-1,X1%+1,Y2%-1,X1%+1,Y1%+1
10340 PRINT "^C1^G!"; : PLOT X1%,Y1%,X2%,Y2%
10350 REM INTERIOR
10360 IF (D2% AND 1)=0 THEN PRINT CHR$(21);CHR$(27);"OA3"; : RETURN
10370 X1%=PX%(QPT%)+2 : Y1%=PY%(QPT%)+2 : PQ%=ABS(PTYPE%(QPT%))
```

Secondary Program Segment
Annotated Source Code Listing

Energy Monitoring and Control System
Man/Machine Interface (EMCS/MMI)
Initial Release - May 1982

```
10380 IF PQ%<=2 THEN X2%=X1%+25 : Y2%=Y1%+25 : GOTO 10450
10390 IF PQ%>=LIND% GOTO 10400 ELSE X2%=X1%+17 : Y2%=Y1%+17 : GOTO 10450
10400 IF PQ%<17 THEN X2%=X1%+23 : Y2%=Y1%+13 : GOTO 10450
10410 IF PQ%=18 THEN X2%=X1%+41 : Y2%=Y1%+54 : GOTO 10450
10420 IF PQ%=20 THEN X2%=X1%+71 : Y2%=Y1%+58 : GOTO 10450
10430 IF PQ%=22 THEN X2%=X1%+23 : Y2%=Y1%+22 : GOTO 10450
10440 IF PQ%<>23 GOTO 10750 ELSE X2%=X1%+3 : Y2%=Y1%+15
10450 IF (D3% AND 2)<>0 THEN PRINT ""C2"; ELSE PRINT ""C0";
10460 PRINT ""F"G+"; : PLOT X1%,Y1%,X2%,Y2% : PRINT ""L";
10470 IF PQ%=23 THEN PRINT ""C1"G!"; : PLOT X1%,Y1%,X2%,Y2%
10490 PRINT CHR$(21) : IF SL%=0 THEN PRINT CHR$(27);"0A3"; : RETURN
10500 Y3%=Y2%-((Y2%-Y1%-9)/2) : L%=SL%
10510 IF (PQ%=1) OR (PQ%=2) OR (PQ%=18) THEN Y3%=Y2%-((Y2%-Y1%-19)/2) : L%=3
10520 IF PQ%=18 THEN L%=2
10530 X3%=X1%+(X2%-X1%-(6*L%+2))/2+1 : TM$=LEFT$(SI$(PQ%),L%)
10550 IF PQ%<LIND% THEN PRINT ""M"C0"N"C3"; : GOTO 10580
10570 PRINT ""C1"; : IF (D3% AND 2)<>0 THEN PRINT ""M"C2"N"; ELSE PRINT ""M"C0"N";
10580 PRINT ""K"U"; : PLOT X3%,Y3% : PRINT TM$ : X3%=X3%+6*L%
10600 IF ((D3% AND 2)=0) OR (PQ%<LI%) THEN PRINT ""C0"; ELSE PRINT ""C2";
10610 PRINT ""F"G+"; : PLOT X3%,Y3%,X3%+1,Y3%-9 : PRINT ""L";CHR$(21);""C3";
10630 IF (PQ%<>1) AND (PQ%<>2) THEN PRINT ""C1"; : GOTO 10650
10640 L%=1 : GOTO 10660
10650 IF PQ%=18 THEN L%=6 ELSE GOTO 10740
10660 REM LINE2
10680 Y3%=Y3%-10 : X3%=X1%+(X2%-X1%-(6*L%+2))/2+1
10700 TM$=RIGHT$(SI$(PQ%),L%) : PRINT ""K"U"; : PLOT X3%,Y3% : PRINT TM$
10720 X3%=X3%+6*L% : IF ((D3% AND 2)=0)OR(PQ%<LI%) THEN PRINT ""C0"; ELSE PRINT ""C2";
10730 PRINT ""F"G+"; : PLOT X3%,Y3%,X3%+1,Y3%-9 : PRINT ""L"; CHR$(21)
10740 PRINT CHR$(27);"0A3"; : RETURN
10750 REM ODD
10760 IF (D3% AND 2)<>0 THEN PRINT ""C2"; ELSE PRINT ""C0";
10770 IF PQ%=21 GOTO 10920
10790 X3%=X1%+9 : Y3%=Y1%+9 : IF PQ%=17 GOTO 10830
10800 REM PUMP
10810 IF QPT%<>84 THEN X4%=X3%+12 ELSE X4%=X3%-12
10820 PRINT ""F"G+"; : PLOT X3%,Y3%+2,X4%,Y3%+11
10830 REM PUMP/TC
10840 PRINT ""F"G*"; : PLOT X3%,Y3%,9 : PRINT ""L";CHR$(21)
10860 IF PQ%<>17 THEN PRINT CHR$(27);"0A3"; : RETURN
10880 X4%=X3%-6 : Y4%=Y3%+5 : IF (D3% AND 2)<>0 THEN PRINT ""M"C2"; ELSE PRINT ""M"C0";
10890 PRINT ""N"C1"K"U"; : PLOT X4%,Y4% : PRINT SID$(PQ%)
10900 IF (D3% AND 2)<>0 THEN PRINT ""C2"; ELSE PRINT ""C0";
```

Secondary Program Segment
Annotated Source Code Listing

Energy Monitoring and Control System
Man/Machine Interface (EMCS/MMI)
Initial Release - May 1982

```
10910 PRINT "^F^G+"; : PLOT X4%+11,Y4%,X4%+12,Y4%-9 : PRINT "^L";CHR$(21); CHR$(27);"OA3"; : RETURN
10920 REM TOWER
10930 X2%=X1%+25 : X3%=X2%+6 : X4%=X1%-6 : Y2%=Y1%+32
10940 IF (D3% AND 2)=0 THEN PRINT "^CO^1"; ELSE PRINT "^M^C2^N^C2";
10950 PRINT "^G("; : PLOT X1%,Y1%,X2%,Y1%,X3%,Y2%,X4%,Y2%,X1%,Y1%
10960 IF (D3% AND 2)=0 THEN PRINT "^M^1^CO^N^1^CO";CHR$(14); ELSE PRINT "^M^C2^2^N^C2";
10970 PRINT "^K^U"; : PLOT X1%+5,Y1%+5 : PRINT "^>4";
10980 IF (D3% AND 2)=0 THEN PRINT "^2^M^2^CO^ : 8";CHR$(12);"^ : F^CO^N^2";CHR$(15);
10990 PRINT "^L";CHR$(21);CHR$(27);"OA3"; : RETURN  .
11000 REM ALARM
11020 PRINT CHR$(27);"OA1"; : X1%=5 : X2%=118 : Y1%=457 : Y2%=506
11030 IF D1%=3 THEN PRINT "^CO"; ELSE PRINT "^1^C4";
11040 PRINT "^F^G+"; : PLOT X1%,Y1%,X2%,Y2% : PRINT "^2^L";CHR$(21);
11050 IF D1%=3 GOTO 11070
11060 PRINT "^M^1^C4^N^2^C7^X3,^Y4,^K^U"; : PLOT 17,501 : PRINT "ALARM";
11070 PRINT "^X1,^Y1,^M^CO^2^N";CHR$(21);CHR$(27);"OA3"; : RETURN
11100 REM DSCHED ==========================================================
11110 FOR I%=1 TO NSE% : SCHTMP$(I%)=S$(QSCH%,I%) : NEXT I%          ' Set up temporary schedule storage.
11130 DOS "DRAW SCHLAY" : PRINT CHR$(27);"OA1";CHR$(12);CHR$(27);"W"; ' Display static image of schedule layout.
11135 PRINT "^U"; : PLOT ((512-(6*LEN(TT$(7,QSCH%))))*2),VPS%(1,0) : PRINT "^C5^X1,^Y2,";TT$(7,QSCH%);"^Y1,";
                                                                     ' Display schedule title on screen.
11140 FOR I%=1 TO NSE%                                               ' Loop to print schedule entries on layout.
11150 PRINT "^U"; : PLOT VPS%(0,I%),VPS%(1,I%) : IF I%<6 THEN PRINT "^C1"; ELSE PRINT "^C3";
                                                                     ' Display first 5 entries in blue and the rest
                                                                       in cyan.
11170 IF SCHTMP$(I%)="NA" THEN PRINT " ";SCHTMP$(I%) ELSE PRINT SCHTMP$(I%)
11180 NEXT I%
11191 PRINT CHR$(27);"OA3"; : MDEX%=50 : GOSUB 7100 : MDEX%=19 : GOSUB 7100
                                                                     ' Move to text window and display cues for
                                                                       operator.
11192 TP%=QFCN% : QFCN%=9 : D1%=0 : D2%=1 : D3%=0 : GOSUB 9200 : QFCN%=TP% : RETURN
                                                                     ' Deselect function key. - Done.
11300 REM MSCHED ==========================================================
11310 I%=6 : X1%=0 : Y1%=0 : J%=0 : K%=0                             ' Initialize variables.
11320 IF (XTUCH%>=VPS%(0,I%)) AND (XTUCH%<=VPS%(0,I%)+24) THEN X1%=VPS%(0,I%) : J%=I%
                                                                     ' Search for selected point.
11330 IF (YTUCH%>=VPS%(1,I%)-20) AND (YTUCH%<=VPS%(1,I%)+10) THEN Y1%=VPS%(1,I%) : K%=I%
11340 IF (J%=K%) AND (J%<>0) GOTO 11360                              ' Has point been found.
11350 IF I%<25 THEN I%=I%+1 : GOTO 11320 ELSE MDEX%=53 : GOSUB 7100 : GOTO 11780
                                                                     ' NO, then check next point or display error.
11360 PRINT CHR$(27);"OA1";"^C6^U"; : PLOT X1%,Y1%                   ' Move cursor to point.
11370 IF  SCHTMP$(K%)="NA" THEN  PRINT " ";SCHTMP$(K%) ELSE PRINT SCHTMP$(K%)
```

Secondary Program Segment
Annotated Source Code Listing

Energy Monitoring and Control System
Man/Machine Interface (EMCS/MMI)
Initial Release - May 1982

```
                                                                  ' Highlight in yellow.
11380 PRINT CHR$(27);"OA3"; : MDEX%=10 : GOSUB 7100               ' Request operator input.
11390 ZIN$="" : GOSUB 200                                         ' Get keyboard input.
11400 IF LEN(ZIN$)=0 GOTO 11670                                   ' Check for invalid input.
11410 IF K%>21 GOTO 11510                                         ' Go check different set of values.
11420 IF ZIN$="NA" GOTO 11690
11430 IF LEN(ZIN$)<>4 GOTO 11670                                  ' Checks for invalid input on occupancy schedule.
11440 I%=1
11450 TP%=INSTR("0123456789",MID$(ZIN$,I%,1))                     ' Loop and check each input character for valid
                                                                    and resonable value.
11460 IF TP%=0 GOTO 11670
11470 IF (I%=1) AND (TP%>3) GOTO 11670
11480 IF (VAL(LEFT$(ZIN$,1))=2) AND (I%=2) AND (TP%>4) GOTO 11670
11490 IF (I%=3) AND (TP%>6) GOTO 11670
11500 I%=I%+1 : IF I%<=4 GOTO 11450 ELSE GOTO 11690
11510 IF K%<>22 GOTO 11530                                        ' Check season input for valid value.
11520 IF (ZIN$="S")OR(ZIN$="W") GOTO 11690 ELSE GOTO 11670
11530 IF K%<>25 GOTO 11610
11540 IF LEN(ZIN$)<>4 GOTO 11670                                  ' Check multiplier input.
11550 I%=1
11560 TP%=INSTR("0123456789.",MID$(ZIN$,I%,1))                    ' Loop and check each input character for an
                                                                    in-range value.
11570 IF TP%=0 GOTO 11670
11580 IF (I%=2) AND (TP%<>11) GOTO 11670
11590 IF ((I%=1)OR(I%>=3)) AND (TP%=11) GOTO 11670
11600 I%=I%+1 : IF I%<=4 GOTO 11560 ELSE GOTO 11690
11610 IF RIGHT$(ZIN$,1)<>"F" GOTO 11670                           ' Check HTG and CLG keyboard input.
11620 IF K%<>23 GOTO 11650                                        ' Is cooling input in range.
11630 IF LEN(ZIN$)>3 GOTO 11670
11640 IF VAL(ZIN$)>75 GOTO 11670 ELSE GOTO 11690
11650 IF LEN(ZIN$)>4 GOTO 11670                                   ' Is heating input in range.
11660 IF (VAL(ZIN$)<70)OR(VAL(ZIN$)>120) GOTO 11670 ELSE GOTO 11690
11670 MDEX%=52 : GOSUB 7100 : GOTO 11380                          ' Come here on error in keyboard input.
11680 REM Change value in array and on screen.
11690 PRINT CHR$(27);"OA1";""CO"U"; : PLOT X1%,Y1% : IF  SCHTMP$(K%)="NA" THEN PRINT " ";SCHTMP$(K%) ELSE PRINT
        SCHTMP$(K%)                                               ' Black out previous value on screen.
11710 REM Update Schedule.
11720 SCHTMP$(K%)=ZIN$ : PRINT ""C3"U"; : PLOT X1%,Y1% : IF SCHTMP$(K%)="NA" THEN PRINT " ";SCHTMP$(K%) ELSE PRINT
        SCHTMP$(K%)                                               ' Print new value on CRT in cyan.
11730 PRINT CHR$(27);"OA3";                                       ' Move back to the text window.
11740 IF K%<>22 GOTO 11780                                        ' Update schedule values depending on season.
11750 IF SCHTMP$(22)="S" GOTO 11770
```

```
11760 SCHTMP$(1)="+45F" : SCHTMP$(3)="70.2F" : SCHTMP$(4)="64.7F" : SCHTMP$(5)="HTG" : GOTO 11780
                                                                   ' (winter values)
11770 SCHTMP$(1)="+8.9F" : SCHTMP$(3)="76.3F" : SCHTMP$(4)="87.1F" : SCHTMP$(5)="CLG"
                                                                   ' (summer values)
11780 MDEX$=50 : GOSUB 7100 : MDEX$=19 : GOSUB 7100 : RETURN      ' Done.
```

APPENDIX F

Database Listing

F-1 Real-Time Database

F-2 Simulation Arrays

F-1

Real-Time Database

Table Of Real-Time Device Data Base Values -- PAGE 1

Variable Name

DEVICE#	ADPT%	ALARMVL%	ASTAT%	DSTAT%	HILIM	LOWLIM
1	1	0	0	0	0.0	0.0
2	1	0	0	0	0.0	0.0
3	1	0	0	0	0.0	0.0
4	1	0	0	0	0.0	0.0
5	1	0	0	0	0.0	0.0
6	1	0	0	0	0.0	0.0
7	1	0	0	0	0.0	0.0
8	0	0	0	0	140.0	100.0
9	0	0	0	0	180.0	140.0
10	0	0	0	0	15.0	5.0
11	1	0	0	0	0.0	0.0
12	1	0	0	0	0.0	0.0
13	1	0	0	0	0.0	0.0
14	1	0	0	0	0.0	0.0
15	1	0	0	0	0.0	0.0
16	0	0	0	0	180.0	140.0
17	0	0	0	0	71.0	65.0
18	0	0	0	0	71.0	65.0
19	0	0	0	0	75.0	61.0
20	1	0	0	0	0.0	0.0
21	0	0	0	0	80.0	32.0
22	0	0	0	0	75.0	32.0
23	0	0	0	0	45.0	20.0
24	1	0	0	0	0.0	0.0

Table Of Real-Time Device Data Base Values -- PAGE 2

Variable Name

DEVICE#	LPNT%	PHIER%	PTYPE%	PX%	PY%	SPT	SSED%
1	1	0	0	319	402	0.0	0
2	1	1	-1	319	360	0.0	0
3	1	2	11	226	309	0.0	0
4	1	2	-4	226	178	0.0	1
5	1	4	-16	223	210	0.0	1
6	1	5	-19	226	241	0.0	1
7	1	6	15	223	276	0.0	0
8	1	2	13	267	182	120.0	0
9	1	2	13	267	272	160.0	0
10	1	2	7	286	312	10.0	0
11	1	2	11	324	328	0.0	0
12	1	2	-4	324	202	0.0	1
13	1	12	-16	321	234	0.0	1
14	1	13	-19	324	265	0.0	1
15	1	14	15	321	300	0.0	0
16	1	2	13	380	298	160.0	0
17	1	2	-3	421	340	68.0	0
18	1	17	17	444	339	68.0	0
19	1	2	13	478	320	68.0	0
20	1	1	-2	111	336	0.0	0
21	1	20	13	76	368	40.0	0
22	1	20	13	18	208	50.0	0
23	1	20	7	36	180	35.0	0
24	1	20	11	60	274	0.0	0

Table Of Real-Time Device Data Base Values -- PAGE 3

	Variable Name					
DEVICE#	ADPT$	ALARMVL$	ASTAT$	DSTAT$	HILIM	LOWLIM
25	1	0	0	0	0.0	0.0
26	1	0	0	0	0.0	0.0
27	0	0	0	0	175.0	150.0
28	0	0	0	0	175.0	150.0
29	1	0	0	0	0.0	0.0
30	1	0	0	0	0.0	0.0
31	1	0	0	0	0.0	0.0
32	1	0	0	0	0.0	0.0
33	0	0	0	0	100.0	32.0
34	0	0	0	0	80.0	10.0
35	1	0	0	0	0.0	0.0
36	0	0	0	0	20.0	10.0
37	0	0	0	0	100.0	80.0
38	0	0	0	0	75.0	65.0
39	1	0	0	0	0.0	0.0
40	0	0	0	0	85.0	60.0
41	0	0	0	0	85.0	60.0
42	1	0	0	0	0.0	0.0
43	0	0	0	0	75.0	25.0
44	1	0	0	0	0.0	0.0
45	1	0	0	0	0.0	0.0
46	1	0	0	0	0.0	0.0
47	0	0	0	0	60.0	50.0
48	0	0	0	0	50.0	40.0

Table Of Real-Time Device Data Base Values -- PAGE 4

Variable Name

DEVICE#	LPNT%	PHIER%	PTYPE%	PX%	PY%	SPT	SSED%
25	1	20	-4	76	302	0.0	1
26	1	25	18	65	164	0.0	1
27	1	20	-3	115	266	160.0	0
28	1	27	17	92	265	160.0	0
29	1	20	9	118	194	0.0	0
30	1	20	8	118	170	0.0	0
31	2	0	0	277	400	0.0	0
32	2	31	-2	277	344	0.0	0
33	2	32	13	223	398	85.0	0
34	2	32	14	223	368	52.0	0
35	2	32	11	195	342	0.0	0
36	2	32	10	217	310	15.0	0
37	2	32	13	271	308	90.0	0
38	2	32	13	291	272	70.0	0
39	2	32	11	333	308	0.0	0
40	2	32	-3	389	180	70.0	0
41	2	40	-17	411	180	70.0	0
42	2	41	21	457	274	0.0	1
43	2	32	12	389	308	50.0	0
44	2	32	11	461	230	0.0	0
45	2	31	-1	69	354	0.0	0
46	2	45	11	65	292	0.0	0
47	2	45	13	84	322	55.0	0
48	2	45	13	129	292	45.0	0

Variable Name

DEVICE#	ADPT%	ALARMVL%	ASTAT%	DSTAT%	HILIM	LOWLIM
49	1	0	0	0	0.0	0.0
50	1	0	0	0	0.0	0.0
51	1	0	0	0	0.0	0.0
52	1	0	0	0	0.0	0.0
53	1	0	0	0	0.0	0.0
54	1	0	0	0	0.0	0.0
55	1	0	0	0	0.0	0.0
56	1	0	0	0	0.0	0.0
57	0	0	0	0	48.0	43.0
58	0	0	0	0	48.0	43.0
59	1	1	0	0	0.0	0.0
60	1	0	0	0	0.0	0.0
61	1	0	0	0	0.0	0.0
62	0	0	0	0	80.0	65.0
63	0	0	0	0	80.0	20.0
64	1	1	0	0	0.0	0.0
65	1	0	0	0	0.0	0.0
66	1	0	0	0	0.0	0.0
67	1	0	0	0	0.0	0.0
68	0	0	0	0	58.0	52.0
69	0	0	0	0	58.0	52.0
70	0	0	0	0	100.0	0.0
71	0	0	0	0	65.0	45.0
72	0	0	0	0	70.0	50.0

Table Of Real-Time Device Data Base Values -- PAGE 6

Variable Name

DEVICE#	LPNT%	PHIER%	PTYPE%	PX%	PY%	SPT	SSED%
49	2	45	-4	65	156	0.0	1
50	2	49	-16	62	188	0.0	1
51	2	49	-16	330	214	0.0	1
52	2	50	-19	65	224	0.0	1
53	2	51	-19	333	246	0.0	1
54	2	52	15	62	264	0.0	0
55	2	53	15	330	279	0.0	0
56	2	50	20	169	236	0.0	1
57	2	45	-3	126	342	45.0	0
58	2	57	17	149	342	45.0	0
59	2	45	5	173	310	0.0	0
60	3	0	0	240	416	0.0	0
61	3	60	-1	288	386	0.0	0
62	3	61	13	112	338	72.0	0
63	3	61	14	160	338	50.0	0
64	3	61	11	183	278	0.0	0
65	3	61	11	229	356	0.0	0
66	3	61	11	265	276	0.0	0
67	3	61	-4	328	362	0.0	1
68	3	61	-3	380	292	55.0	0
69	3	68	-17	358	292	55.0	0
70	3	61	12	398	231	50.0	0
71	3	61	13	428	292	55.0	0
72	3	61	13	486	280	60.0	0

Table Of Real-Time Device Data Base Values -- PAGE 7

Variable Name

DEVICE#	ADPT%	ALARMVL%	ASTAT%	DSTAT%	HILIM	LOWLIM
73	1	0	0	0	0.0	0.0
74	0	0	0	0	78.0	68.0
75	0	0	0	0	70.0	30.0
76	0	0	0	0	110.0	50.0
77	0	0	0	0	100.0	60.0
78	0	0	0	0	100.0	60.0
79	1	0	0	0	0.0	0.0
80	1	0	0	0	0.0	0.0
81	1	0	0	0	0.0	0.0
82	1	0	0	0	0.0	0.0
83	1	0	0	0	0.0	0.0
84	1	0	0	0	0.0	0.0
85	1	0	0	0	0.0	0.0
86	0	0	0	0	85.0	50.0
87	1	0	0	0	0.0	0.0
88	0	0	0	0	100.0	32.0
89	0	0	0	0	100.0	10.0
90	1	0	0	0	0.0	0.0
91	1	0	0	0	0.0	0.0
92	1	0	0	0	0.0	0.0
93	1	0	0	0	0.0	0.0

Table Of Real-Time Device Data Base Values -- PAGE 8

Variable Name

DEVICE#	LPNT%	PHIER%	PTYPE%	PX%	PY%	SPT	SSED%
73	3	60	-2	240	138	0.0	0
74	3	73	13	486	144	73.0	0
75	3	73	14	452	144	50.0	0
76	3	73	13	410	164	80.0	0
77	3	73	-3	378	172	80.0	0
78	3	77	-17	356	172	80.0	0
79	3	73	-4	304	172	0.0	1
80	3	79	-16	262	196	0.0	1
81	3	80	-19	265	225	0.0	1
82	3	81	15	262	253	0.0	0
83	3	79	-16	226	266	0.0	1
84	3	83	-19	229	296	0.0	1
85	3	84	15	226	328	0.0	0
86	3	73	13	200	196	67.0	0
87	3	73	6	14	263	0.0	0
88	3	73	13	48	178	85.0	0
89	3	73	14	48	140	70.0	0
90	3	67	22	326	322	0.0	1
91	3	69	23	344	256	0.0	1
92	3	78	23	344	208	0.0	1
93	3	61	15	180	252	0.0	0

F-2

Simulation Arrays

Table Of Real-Time Device Simulation Values -- PAGE 1

DEVICE#	Simulation Time Period 1	2	3	4	5	6	7
1	1.0	1.0	1.0	1.0	1.0	1.0	1.0
2	1.0	1.0	1.0	1.0	1.0	1.0	1.0
3	1.0	1.0	1.0	1.0	1.0	1.0	1.0
4	1.0	1.0	1.0	1.0	1.0	1.0	1.0
5	1.0	1.0	1.0	1.0	1.0	1.0	1.0
6	1.0	1.0	1.0	1.0	1.0	1.0	1.0
7	1.0	1.0	0.0	1.0	0.0	1.0	1.0
8	110.0	110.0	110.0	110.0	110.0	110.0	110.0
9	165.0	165.0	165.0	165.0	165.0	165.0	165.0
10	10.0	10.0	10.0	0.0	10.0	0.0	10.0
11	1.0	1.0	1.0	1.0	1.0	1.0	1.0
12	1.0	1.0	1.0	1.0	1.0	1.0	1.0
13	1.0	1.0	1.0	1.0	1.0	1.0	1.0
14	1.0	1.0	1.0	1.0	1.0	1.0	1.0
15	1.0	1.0	1.0	1.0	1.0	1.0	1.0
16	160.0	160.0	160.0	160.0	160.0	160.0	160.0
17	68.0	68.0	68.0	68.0	68.0	68.0	68.0
18	68.0	68.0	68.0	68.0	68.0	68.0	68.0
19	65.0	65.0	68.0	68.0	68.0	68.0	71.0
20	1.0	1.0	1.0	1.0	1.0	1.0	1.0
21	32.0	32.0	35.0	35.0	35.0	35.0	35.0
22	50.0	50.0	50.0	50.0	50.0	50.0	50.0
23	35.0	35.0	35.0	35.0	35.0	35.0	35.0
24	1.0	1.0	1.0	1.0	1.0	1.0	1.0

Table Of Real-Time Device Simulation Values -- PAGE 2

DEVICE#	Simulation Time Period 8	9	10	11	12	13	14
1	1.0	1.0	1.0	1.0	1.0	1.0	1.0
2	1.0	1.0	1.0	1.0	1.0	1.0	1.0
3	1.0	1.0	1.0	1.0	1.0	1.0	1.0
4	1.0	1.0	1.0	1.0	1.0	1.0	1.0
5	1.0	1.0	1.0	1.0	1.0	1.0	1.0
6	1.0	1.0	1.0	1.0	1.0	1.0	1.0
7	1.0	1.0	1.0	1.0	1.0	1.0	1.0
8	110.0	110.0	110.0	115.0	115.0	115.0	115.0
9	165.0	165.0	165.0	160.0	160.0	160.0	160.0
10	10.0	10.0	10.0	10.0	10.0	10.0	10.0
11	1.0	1.0	1.0	1.0	1.0	1.0	1.0
12	1.0	1.0	1.0	1.0	1.0	1.0	1.0
13	1.0	1.0	1.0	1.0	1.0	1.0	1.0
14	1.0	1.0	1.0	1.0	1.0	1.0	1.0
15	1.0	1.0	1.0	1.0	1.0	1.0	1.0
16	160.0	135.0	135.0	155.0	155.0	155.0	155.0
17	68.0	68.0	68.0	68.0	68.0	68.0	68.0
18	68.0	68.0	68.0	68.0	68.0	68.0	68.0
19	71.0	72.0	72.0	72.0	72.0	72.0	72.0
20	1.0	1.0	1.0	1.0	1.0	1.0	1.0
21	35.0	38.0	38.0	38.0	38.0	41.0	41.0
22	50.0	50.0	50.0	50.0	50.0	50.0	50.0
23	35.0	35.0	35.0	35.0	35.0	35.0	35.0
24	1.0	1.0	1.0	1.0	1.0	1.0	1.0

Table Of Real-Time Device Simulation Values -- PAGE 3

DEVICE#	Simulation Time Period 15	16	17	18	19	20	21
1	1.0	1.0	1.0	1.0	1.0	1.0	1.0
2	1.0	1.0	1.0	1.0	1.0	1.0	1.0
3	1.0	1.0	1.0	1.0	1.0	1.0	1.0
4	1.0	1.0	1.0	1.0	1.0	1.0	1.0
5	1.0	1.0	1.0	1.0	1.0	1.0	1.0
6	1.0	1.0	1.0	1.0	1.0	1.0	1.0
7	1.0	1.0	1.0	1.0	1.0	1.0	1.0
8	115.0	115.0	120.0	120.0	115.0	115.0	115.0
9	160.0	160.0	155.0	155.0	160.0	160.0	160.0
10	10.0	10.0	10.0	10.0	10.0	10.0	10.0
11	1.0	1.0	1.0	1.0	1.0	1.0	1.0
12	1.0	1.0	1.0	1.0	1.0	1.0	1.0
13	1.0	1.0	1.0	1.0	1.0	1.0	1.0
14	1.0	1.0	1.0	1.0	1.0	1.0	1.0
15	1.0	1.0	1.0	1.0	1.0	1.0	1.0
16	155.0	155.0	150.0	150.0	155.0	155.0	155.0
17	68.0	68.0	68.0	68.0	68.0	68.0	68.0
18	68.0	68.0	68.0	68.0	68.0	68.0	68.0
19	73.0	73.0	74.0	74.0	74.0	74.0	73.0
20	1.0	1.0	1.0	1.0	1.0	1.0	1.0
21	41.0	41.0	45.0	45.0	42.0	42.0	42.0
22	50.0	50.0	50.0	50.0	50.0	50.0	50.0
23	35.0	35.0	35.0	35.0	35.0	35.0	35.0
24	1.0	1.0	1.0	1.0	1.0	1.0	1.0

Table Of Real-Time Device Simulation Values -- PAGE 4

	Simulation Time Period						
DEVICE#	22	23	24	25	26	27	28
1	1.0	1.0	1.0	1.0	1.0	1.0	1.0
2	1.0	1.0	1.0	1.0	1.0	1.0	1.0
3	1.0	1.0	1.0	1.0	1.0	1.0	1.0
4	1.0	1.0	1.0	1.0	1.0	1.0	1.0
5	1.0	1.0	1.0	1.0	1.0	1.0	1.0
6	1.0	1.0	1.0	1.0	1.0	1.0	1.0
7	1.0	1.0	1.0	1.0	1.0	1.0	1.0
8	115.0	115.0	115.0	115.0	115.0	110.0	110.0
9	160.0	165.0	165.0	165.0	165.0	165.0	165.0
10	10.0	10.0	10.0	10.0	10.0	10.0	10.0
11	1.0	1.0	1.0	1.0	1.0	1.0	1.0
12	1.0	1.0	1.0	1.0	1.0	1.0	1.0
13	1.0	1.0	1.0	1.0	1.0	1.0	1.0
14	1.0	1.0	1.0	1.0	1.0	1.0	1.0
15	1.0	1.0	1.0	1.0	1.0	1.0	1.0
16	155.0	160.0	160.0	160.0	160.0	160.0	160.0
17	68.0	68.0	68.0	68.0	68.0	68.0	68.0
18	68.0	68.0	68.0	68.0	68.0	68.0	68.0
19	73.0	70.0	70.0	70.0	70.0	68.0	68.0
20	1.0	1.0	1.0	1.0	1.0	1.0	1.0
21	42.0	39.0	39.0	39.0	39.0	35.0	35.0
22	50.0	50.0	50.0	50.0	50.0	50.0	50.0
23	35.0	35.0	35.0	35.0	35.0	35.0	35.0
24	1.0	1.0	1.0	1.0	1.0	1.0	1.0

Table Of Real-Time Device Simulation Values -- PAGE 5

DEVICE#	1	2	3	4	5	6	7
	Simulation Time Period						
25	1.0	1.0	1.0	1.0	1.0	1.0	1.0
26	1.0	1.0	1.0	1.0	1.0	1.0	1.0
27	165.0	165.0	165.0	165.0	165.0	165.0	165.0
28	165.0	165.0	165.0	165.0	165.0	165.0	165.0
29	1.0	1.0	1.0	1.0	1.0	1.0	1.0
30	1.0	1.0	1.0	1.0	1.0	1.0	1.0
31	1.0	1.0	1.0	1.0	1.0	1.0	1.0
32	1.0	1.0	1.0	1.0	1.0	1.0	1.0
33	75.0	75.0	75.0	75.0	78.0	78.0	78.0
34	65.0	65.0	65.0	65.0	62.0	62.0	62.0
35	1.0	1.0	1.0	1.0	1.0	1.0	1.0
36	15.0	15.0	15.0	15.0	15.0	15.0	15.0
37	85.0	85.0	85.0	85.0	88.0	88.0	88.0
38	66.0	66.0	66.0	66.0	69.0	69.0	69.0
39	1.0	1.0	1.0	1.0	1.0	1.0	1.0
40	70.0	70.0	70.0	70.0	70.0	70.0	70.0
41	70.0	70.0	70.0	70.0	70.0	70.0	70.0
42	1.0	1.0	1.0	1.0	1.0	1.0	1.0
43	50.0	50.0	45.0	45.0	40.0	40.0	35.0
44	1.0	1.0	1.0	1.0	1.0	1.0	1.0
45	1.0	1.0	1.0	1.0	1.0	1.0	1.0
46	1.0	1.0	1.0	1.0	1.0	1.0	1.0
47	55.0	55.0	55.0	55.0	55.0	55.0	55.0
48	45.0	45.0	45.0	45.0	45.0	45.0	45.0

Table Of Real-Time Device Simulation Values -- PAGE 6

DEVICE#	Simulation Time Period 8	9	10	11	12	13	14
25	1.0	1.0	1.0	1.0	1.0	1.0	1.0
26	1.0	1.0	1.0	1.0	1.0	1.0	1.0
27	165.0	165.0	165.0	165.0	165.0	165.0	165.0
28	165.0	165.0	165.0	165.0	165.0	165.0	165.0
29	1.0	1.0	1.0	1.0	1.0	1.0	1.0
30	1.0	1.0	1.0	1.0	1.0	1.0	1.0
31	1.0	1.0	1.0	1.0	1.0	1.0	1.0
32	1.0	1.0	1.0	1.0	1.0	1.0	1.0
33	78.0	81.0	81.0	81.0	81.0	85.0	85.0
34	62.0	59.0	59.0	59.0	59.0	55.0	55.0
35	1.0	1.0	1.0	1.0	1.0	1.0	1.0
36	15.0	15.0	15.0	15.0	15.0	15.0	15.0
37	88.0	91.0	91.0	91.0	91.0	95.0	95.0
38	69.0	72.0	72.0	72.0	72.0	80.0	80.0
39	1.0	1.0	1.0	1.0	1.0	1.0	1.0
40	70.0	70.0	70.0	70.0	70.0	70.0	70.0
41	70.0	70.0	70.0	70.0	70.0	70.0	70.0
42	1.0	1.0	1.0	1.0	1.0	1.0	1.0
43	35.0	35.0	35.0	30.0	30.0	70.0	70.0
44	1.0	1.0	1.0	1.0	1.0	1.0	1.0
45	1.0	1.0	1.0	1.0	1.0	1.0	1.0
46	1.0	1.0	1.0	1.0	1.0	1.0	1.0
47	55.0	55.0	55.0	55.0	55.0	55.0	55.0
48	45.0	45.0	45.0	45.0	45.0	45.0	45.0

Table Of Real-Time Device Simulation Values -- PAGE 7

DEVICE#	Simulation Time Period 15	16	17	18	19	20	21
25	1.0	1.0	1.0	1.0	1.0	1.0	1.0
26	1.0	1.0	1.0	1.0	1.0	1.0	1.0
27	165.0	165.0	165.0	165.0	165.0	165.0	165.0
28	165.0	165.0	165.0	165.0	165.0	165.0	165.0
29	1.0	1.0	1.0	1.0	1.0	1.0	1.0
30	1.0	1.0	1.0	1.0	1.0	1.0	1.0
31	1.0	1.0	1.0	1.0	1.0	1.0	1.0
32	1.0	1.0	1.0	1.0	1.0	1.0	1.0
33	88.0	88.0	88.0	88.0	92.0	92.0	92.0
34	52.0	52.0	52.0	52.0	50.0	50.0	50.0
35	1.0	1.0	1.0	1.0	1.0	1.0	1.0
36	15.0	15.0	2.0	2.0	15.0	15.0	15.0
37	95.0	95.0	95.0	95.0	91.0	91.0	91.0
38	73.0	73.0	74.0	74.0	74.0	74.0	74.0
39	1.0	1.0	1.0	1.0	1.0	1.0	1.0
40	70.0	70.0	70.0	70.0	70.0	70.0	70.0
41	70.0	70.0	70.0	70.0	70.0	70.0	70.0
42	1.0	1.0	1.0	1.0	1.0	1.0	1.0
43	30.0	30.0	25.0	25.0	25.0	25.0	25.0
44	1.0	1.0	1.0	1.0	1.0	1.0	1.0
45	1.0	1.0	1.0	1.0	1.0	1.0	1.0
46	1.0	1.0	1.0	1.0	1.0	1.0	1.0
47	55.0	55.0	59.0	59.0	55.0	55.0	55.0
48	45.0	45.0	45.0	45.0	45.0	45.0	45.0

Table Of Real-Time Device Simulation Values -- PAGE 8

	Simulation Time Period						
DEVICE#	22	23	24	25	26	27	28
25	1.0	1.0	1.0	1.0	1.0	1.0	1.0
26	1.0	1.0	1.0	1.0	1.0	1.0	1.0
27	165.0	165.0	165.0	165.0	165.0	165.0	165.0
28	165.0	165.0	165.0	165.0	165.0	165.0	165.0
29	1.0	1.0	1.0	1.0	1.0	1.0	1.0
30	1.0	1.0	1.0	1.0	1.0	1.0	1.0
31	1.0	1.0	1.0	1.0	1.0	1.0	1.0
32	1.0	1.0	1.0	1.0	1.0	1.0	1.0
33	92.0	88.0	88.0	81.0	81.0	78.0	78.0
34	50.0	54.0	54.0	58.0	58.0	62.0	62.0
35	1.0	1.0	1.0	1.0	1.0	1.0	1.0
36	15.0	15.0	15.0	15.0	15.0	15.0	15.0
37	91.0	88.0	88.0	88.0	88.0	86.0	86.0
38	74.0	73.0	73.0	67.0	67.0	64.0	64.0
39	1.0	1.0	1.0	1.0	1.0	1.0	1.0
40	70.0	70.0	70.0	70.0	70.0	70.0	70.0
41	70.0	70.0	70.0	70.0	70.0	70.0	70.0
42	1.0	1.0	1.0	1.0	1.0	1.0	1.0
43	25.0	30.0	30.0	35.0	35.0	40.0	40.0
44	1.0	1.0	1.0	1.0	1.0	1.0	1.0
45	1.0	1.0	1.0	1.0	1.0	1.0	1.0
46	1.0	1.0	1.0	1.0	1.0	1.0	1.0
47	55.0	55.0	55.0	55.0	55.0	55.0	55.0
48	45.0	45.0	45.0	45.0	45.0	45.0	45.0

DEVICE#	Simulation Time Period 1	2	3	4	5	6	7
49	1.0	1.0	1.0	1.0	1.0	1.0	1.0
50	1.0	1.0	1.0	1.0	1.0	1.0	1.0
51	1.0	1.0	1.0	1.0	1.0	1.0	1.0
52	1.0	1.0	1.0	1.0	1.0	1.0	1.0
53	1.0	1.0	1.0	1.0	1.0	1.0	1.0
54	1.0	1.0	1.0	1.0	1.0	1.0	1.0
55	1.0	1.0	1.0	1.0	1.0	1.0	1.0
56	1.0	1.0	1.0	1.0	1.0	1.0	1.0
57	45.0	45.0	45.0	45.0	45.0	45.0	45.0
58	45.0	45.0	45.0	45.0	45.0	45.0	45.0
59	0.0	0.0	0.0	0.0	0.0	0.0	0.0
60	1.0	1.0	1.0	1.0	1.0	1.0	1.0
61	1.0	1.0	1.0	1.0	1.0	1.0	1.0
62	67.0	67.0	67.0	67.0	70.0	70.0	70.0
63	40.0	40.0	40.0	40.0	40.0	40.0	40.0
64	0.0	0.0	0.0	0.0	0.0	0.0	0.0
65	1.0	1.0	1.0	1.0	1.0	1.0	1.0
66	1.0	1.0	1.0	1.0	1.0	1.0	1.0
67	1.0	1.0	1.0	1.0	1.0	1.0	1.0
68	55.0	55.0	55.0	55.0	55.0	55.0	55.0
69	55.0	55.0	55.0	55.0	55.0	55.0	55.0
70	50.0	50.0	50.0	50.0	50.0	50.0	50.0
71	55.0	55.0	55.0	55.0	55.0	55.0	55.0
72	60.0	60.0	60.0	60.0	60.0	60.0	60.0

Table Of Real-Time Device Simulation Values -- PAGE 10

DEVICE#	Simulation Time Period 8	9	10	11	12	13	14
49	1.0	1.0	1.0	1.0	1.0	1.0	1.0
50	1.0	1.0	1.0	1.0	1.0	1.0	1.0
51	1.0	1.0	1.0	1.0	1.0	1.0	1.0
52	1.0	1.0	1.0	1.0	1.0	1.0	1.0
53	1.0	1.0	1.0	1.0	1.0	1.0	1.0
54	1.0	1.0	1.0	1.0	1.0	1.0	1.0
55	1.0	1.0	1.0	1.0	1.0	1.0	1.0
56	1.0	1.0	1.0	1.0	1.0	1.0	1.0
57	45.0	45.0	45.0	45.0	45.0	45.0	45.0
58	45.0	45.0	45.0	45.0	45.0	45.0	45.0
59	0.0	0.0	0.0	0.0	0.0	0.0	0.0
60	1.0	1.0	1.0	1.0	1.0	1.0	1.0
61	1.0	1.0	1.0	1.0	1.0	1.0	1.0
62	70.0	73.0	73.0	73.0	73.0	75.0	75.0
63	40.0	40.0	40.0	40.0	40.0	40.0	40.0
64	0.0	0.0	0.0	0.0	0.0	0.0	0.0
65	1.0	1.0	1.0	1.0	1.0	1.0	1.0
66	1.0	1.0	1.0	1.0	1.0	1.0	1.0
67	1.0	1.0	1.0	1.0	1.0	1.0	1.0
68	55.0	55.0	55.0	55.0	55.0	55.0	55.0
69	55.0	55.0	55.0	55.0	55.0	55.0	55.0
70	50.0	50.0	50.0	50.0	50.0	50.0	50.0
71	55.0	55.0	55.0	55.0	55.0	55.0	55.0
72	60.0	60.0	60.0	60.0	60.0	60.0	60.0

DEVICE#	15	16	17	18	19	20	21
	Simulation Time Period						
49	1.0	1.0	1.0	1.0	1.0	1.0	1.0
50	1.0	1.0	1.0	1.0	1.0	1.0	1.0
51	1.0	1.0	1.0	1.0	1.0	1.0	1.0
52	1.0	1.0	1.0	1.0	1.0	1.0	1.0
53	1.0	1.0	1.0	1.0	1.0	1.0	1.0
54	1.0	1.0	1.0	1.0	1.0	1.0	1.0
55	1.0	1.0	1.0	1.0	1.0	1.0	1.0
56	1.0	1.0	1.0	1.0	1.0	1.0	1.0
57	45.0	45.0	45.0	45.0	45.0	45.0	45.0
58	45.0	45.0	45.0	45.0	45.0	45.0	45.0
59	0.0	0.0	0.0	0.0	0.0	0.0	0.0
60	1.0	1.0	1.0	1.0	1.0	1.0	1.0
61	1.0	1.0	1.0	1.0	1.0	1.0	1.0
62	75.0	78.0	78.0	78.0	76.0	76.0	74.0
63	40.0	40.0	40.0	40.0	40.0	40.0	40.0
64	0.0	0.0	0.0	0.0	0.0	0.0	0.0
65	1.0	1.0	1.0	1.0	1.0	1.0	1.0
66	1.0	1.0	1.0	1.0	1.0	1.0	1.0
67	1.0	1.0	1.0	1.0	1.0	1.0	1.0
68	55.0	55.0	55.0	55.0	55.0	55.0	55.0
69	55.0	55.0	55.0	55.0	55.0	55.0	55.0
70	50.0	50.0	50.0	50.0	50.0	50.0	10.0
71	55.0	55.0	55.0	55.0	55.0	55.0	55.0
72	60.0	60.0	60.0	60.0	60.0	60.0	80.0

Table Of Real-Time Device Simulation Values -- PAGE 12

	Simulation Time Period						
DEVICE#	22	23	24	25	26	27	28
49	1.0	1.0	1.0	1.0	1.0	1.0	1.0
50	1.0	1.0	1.0	1.0	1.0	1.0	1.0
51	1.0	1.0	1.0	1.0	1.0	1.0	1.0
52	1.0	1.0	1.0	1.0	1.0	1.0	1.0
53	1.0	1.0	1.0	1.0	1.0	1.0	1.0
54	1.0	1.0	1.0	1.0	1.0	1.0	1.0
55	1.0	1.0	1.0	1.0	1.0	1.0	1.0
56	1.0	1.0	1.0	1.0	1.0	1.0	1.0
57	45.0	45.0	45.0	45.0	45.0	45.0	45.0
58	45.0	45.0	45.0	45.0	45.0	45.0	45.0
59	0.0	0.0	0.0	0.0	0.0	0.0	0.0
60	1.0	1.0	1.0	1.0	1.0	1.0	1.0
61	1.0	1.0	1.0	1.0	1.0	1.0	1.0
62	74.0	71.0	71.0	71.0	71.0	71.0	71.0
63	40.0	40.0	40.0	40.0	40.0	40.0	40.0
64	0.0	0.0	0.0	0.0	0.0	0.0	0.0
65	1.0	1.0	1.0	1.0	1.0	1.0	1.0
66	1.0	1.0	1.0	1.0	1.0	1.0	1.0
67	1.0	1.0	1.0	1.0	1.0	1.0	1.0
68	55.0	55.0	55.0	55.0	55.0	55.0	55.0
69	55.0	50.0	50.0	50.0	50.0	50.0	50.0
70	10.0	10.0	10.0	10.0	10.0	10.0	10.0
71	55.0	55.0	55.0	55.0	55.0	55.0	55.0
72	80.0	60.0	60.0	60.0	60.0	60.0	60.0

Table Of Real-Time Device Simulation Values -- PAGE 13

DEVICE#	Simulation Time Period 1	2	3	4	5	6	7
73	1.0	1.0	1.0	1.0	1.0	1.0	1.0
74	73.0	73.0	73.0	73.0	73.0	73.0	73.0
75	40.0	40.0	40.0	40.0	40.0	40.0	40.0
76	80.0	80.0	80.0	80.0	80.0	80.0	80.0
77	80.0	80.0	80.0	80.0	80.0	80.0	80.0
78	80.0	80.0	80.0	80.0	80.0	80.0	80.0
79	1.0	1.0	1.0	1.0	1.0	1.0	1.0
80	1.0	1.0	1.0	1.0	1.0	1.0	1.0
81	1.0	1.0	1.0	1.0	1.0	1.0	1.0
82	1.0	1.0	1.0	1.0	1.0	1.0	1.0
83	1.0	1.0	1.0	1.0	1.0	1.0	1.0
84	1.0	1.0	1.0	1.0	1.0	1.0	1.0
85	1.0	1.0	1.0	1.0	1.0	1.0	1.0
86	58.0	58.0	60.0	60.0	62.0	62.0	64.0
87	1.0	1.0	1.0	1.0	1.0	1.0	1.0
88	50.0	50.0	55.0	55.0	60.0	60.0	65.0
89	40.0	40.0	40.0	40.0	40.0	40.0	40.0
90	1.0	1.0	1.0	1.0	1.0	1.0	1.0
91	1.0	1.0	1.0	1.0	1.0	1.0	1.0
92	1.0	1.0	1.0	1.0	1.0	1.0	1.0
93	0.0	0.0	0.0	0.0	0.0	0.0	0.0

	Simulation Time Period						
DEVICE#	8	9	10	11	12	13	14
73	1.0	1.0	1.0	1.0	1.0	1.0	1.0
74	73.0	73.0	73.0	73.0	73.0	73.0	73.0
75	40.0	40.0	40.0	40.0	40.0	40.0	40.0
76	80.0	80.0	80.0	80.0	80.0	80.0	80.0
77	80.0	80.0	80.0	80.0	80.0	80.0	80.0
78	80.0	80.0	80.0	80.0	80.0	80.0	80.0
79	1.0	1.0	1.0	1.0	1.0	1.0	1.0
80	1.0	1.0	1.0	1.0	1.0	1.0	1.0
81	1.0	1.0	1.0	1.0	1.0	1.0	1.0
82	1.0	1.0	1.0	1.0	1.0	1.0	1.0
83	1.0	1.0	1.0	1.0	1.0	1.0	1.0
84	1.0	1.0	1.0	1.0	1.0	1.0	1.0
85	1.0	1.0	1.0	1.0	1.0	1.0	1.0
86	64.0	67.0	67.0	70.0	70.0	70.0	75.0
87	1.0	1.0	1.0	1.0	1.0	1.0	1.0
88	65.0	70.0	70.0	75.0	75.0	75.0	80.0
89	40.0	38.0	38.0	38.0	38.0	38.0	35.0
90	1.0	1.0	1.0	1.0	1.0	1.0	1.0
91	1.0	1.0	1.0	1.0	1.0	1.0	1.0
92	1.0	1.0	1.0	1.0	1.0	1.0	1.0
93	0.0	0.0	0.0	0.0	0.0	0.0	0.0

	Simulation Time Period						
DEVICE#	15	16	17	18	19	20	21
73	1.0	1.0	1.0	1.0	1.0	1.0	1.0
74	73.0	73.0	73.0	73.0	73.0	73.0	77.0
75	40.0	40.0	40.0	40.0	40.0	40.0	40.0
76	80.0	80.0	80.0	80.0	80.0	80.0	80.0
77	80.0	80.0	80.0	80.0	80.0	80.0	80.0
78	80.0	80.0	80.0	80.0	80.0	80.0	80.0
79	1.0	1.0	1.0	1.0	1.0	1.0	1.0
80	1.0	1.0	1.0	1.0	1.0	1.0	1.0
81	1.0	1.0	1.0	1.0	1.0	1.0	1.0
82	1.0	1.0	1.0	1.0	1.0	1.0	1.0
83	1.0	1.0	1.0	1.0	1.0	1.0	1.0
84	1.0	1.0	1.0	1.0	1.0	1.0	1.0
85	1.0	1.0	1.0	1.0	1.0	1.0	1.0
86	75.0	80.0	80.0	77.0	77.0	73.0	73.0
87	1.0	1.0	1.0	1.0	1.0	1.0	1.0
88	80.0	85.0	85.0	80.0	80.0	75.0	75.0
89	35.0	35.0	35.0	35.0	35.0	38.0	38.0
90	1.0	1.0	1.0	1.0	1.0	1.0	1.0
91	1.0	1.0	1.0	1.0	1.0	1.0	1.0
92	1.0	1.0	1.0	1.0	1.0	1.0	1.0
93	0.0	0.0	0.0	0.0	0.0	0.0	0.0

Table Of Real-Time Device Simulation Values -- PAGE 16

	Simulation Time Period						
DEVICE#	22	23	24	25	26	27	28
73	1.0	1.0	1.0	1.0	1.0	1.0	1.0
74	77.0	73.0	73.0	73.0	73.0	73.0	73.0
75	40.0	40.0	40.0	40.0	40.0	40.0	40.0
76	80.0	80.0	80.0	80.0	80.0	80.0	80.0
77	80.0	80.0	80.0	80.0	80.0	80.0	80.0
78	80.0	80.0	80.0	80.0	80.0	80.0	80.0
79	1.0	1.0	1.0	1.0	1.0	1.0	1.0
80	1.0	1.0	1.0	1.0	1.0	1.0	1.0
81	1.0	1.0	1.0	1.0	1.0	1.0	1.0
82	1.0	1.0	1.0	1.0	1.0	1.0	1.0
83	1.0	1.0	1.0	1.0	1.0	1.0	1.0
84	1.0	1.0	1.0	1.0	1.0	1.0	1.0
85	1.0	1.0	1.0	0.0	0.0	1.0	1.0
86	73.0	70.0	70.0	60.0	60.0	60.0	60.0
87	1.0	1.0	1.0	1.0	1.0	1.0	1.0
88	75.0	70.0	70.0	60.0	60.0	60.0	60.0
89	38.0	38.0	38.0	40.0	40.0	40.0	40.0
90	1.0	1.0	1.0	1.0	1.0	1.0	1.0
91	1.0	1.0	1.0	1.0	1.0	1.0	1.0
92	1.0	1.0	1.0	1.0	1.0	1.0	1.0
93	0.0	0.0	0.0	0.0	0.0	0.0	0.0